AD 633720

RADC-TR-66-67, Volume I
Final Report

ORBIT DIFFERENTIAL CORRECTION – TRACKING PROGRAM

Volume I – Differential Correction Geocentric Orbit Computations Program

George E. Townsend

TECHNICAL REPORT NO. RADC-TR-66-67
April 1966

Distribution of This Document is Unlimited

Rome Air Development Center
Research and Technology Division
Air Force Systems Command
Griffiss Air Force Base, New York

ORBIT DIFFERENTIAL CORRECTION - TRACKING PROGRAM

Volume I – Differential Correction Geocentric Orbit Computations Program

George E. Townsend

Distribution of This Document is Unlimited

BRACELAND BROTHERS INC., MAY 1966, 103 BOOKS

FOREWORD

The Space and Information Systems Division (S&ID) of North American Aviation, Inc. (NAA) under Contract AF 30(602)-3638 with the Rome Air Development Center (RADC) of the United States Air Force agreed to perform a 10-month study designed to develop digital computer techniques in two areas of interest to the RADC tracking facility. First, a differential correction geocentric orbit computation program for reducing observed data was to be prepared which would operate in a near-optimum manner at the RADC computer center. Secondly, a computational logic which could be utilized in the tracking process for driving the tracking antennae in an open-loop mode was to be prepared. This second program would employ general perturbations theory in the definition of the predicted trajectory. (This latter task is reported in SID 65-1203-3).

This report was prepared as partial documentation of the first task. The contents present the program logic and FØRTRAN listings for the main body of the required program. The remaining portion of the logic for this program is associated with the identification, ordering, and smoothing of the raw data; this information is presented in the discussion of the processor in a separate document (SID 65-1203-2). This document should be reviewed carefully, since the interface between the two portions of the differential corrections orbit computation program is a magnetic tape containing the smoothed data in a compatible format.

The differential corrections program (DCP) is designed in such a manner that residuals associated with any one of the six types of data

1. Range
2. Range-Rate
3. Azimuth and Elevation
4. Range and Range-Rate
5. Range, Azimuth, and Elevation
6. Range-Rate, Azimuth, and Elevation

(defined by differencing the observed data and a computed set evaluated on the best estimate of the trajectory) are minimized in the sense of minimum variance. When this operation is accomplished (recursively), the nominal trajectory is adjusted. Thus, after processing data acquired over some as yet unspecified interval of time, the trajectory will be known to an accuracy sufficient to allow it to be predicted with confidence. At this time, data can be prepared to allow the program developed as the second task in this contract to be activated to acquire more data. In a sense, then, the system would be self-perpetuating.

Complete discussions of the program logic, and operation of the DCP will be presented in the various sections of this document. However, the most informative

discussions for the user will probably be included in the analysis of the sample problem (section 3.). This conclusion is based on the fact that complete descriptions of program input, program operation, and program output for a real problem are included. The remaining portion of the document will be a necessary part of the library of the programmer who wishes to modify or adapt the program to another use and of the individual who wishes to fully comprehend the rationale which is mechanized.

This contract has been managed at NAA S&ID by Mr. J. A. Hill and directed by Mr. G. E. Townsend. Mr. Townsend also designed the rationale for the program, coded the major portion of the logic, performed the preliminary checks of the operation, and prepared this document. Final checkout of the program was accomplished with the aid of Mr. C. C. DeBilzan.

The assistance offered by RADC personnel under the direction of Mr. Gordon Negus (Program Manager) is gratefully acknowledged. RADC Project 4519 applies.

This technical report has been reviewed and is approved.

Approved: Gordon F. Negus
GORDON F. NEGUS
Project Engineer

Approved: Charles A. Strom, Jr.
CHARLES A. STROM, JR.
Acting Chief
Communications Division

FOR THE COMMANDER: Irving J. Gabelman
IRVING J. GABELMAN
Chief, Advanced Studies Group

ABSTRACT

This document presents the formulation, computational logic and coding information developed for the purpose of effecting the definition of geocentric satellite orbits. The rationale for this process is constructed around the recursive minimum variance data filter developed by R.E. Kalman and a specially prepared magnetic tape generated in the preprocessor (SID 65 1203-2).

The trajectory portion of the program is formulated in the Encke manner and includes perturbing accelerations resulting from the first 3 harmonics of the Earth's potential function, atmospheric drag, solar radiation pressure, and solar and lunar gravitation. These accelerations are integrated via an uncorrected Gauss-Jackson routine started with a fourth order Runge-Kutta process.

TABLE OF CONTENTS

Page

LIST OF SUBROUTINES AND THEIR FUNCTIONS

SECTION	ROUTINE	DEFINITION
2.1	MAIN	This routine mechanizes the subroutines and lower order driven routines designed to generate the trajectory of a satellite vehicle by a differential corrections process.
2.2.1	INPUT	INPUT provides all of the data (except the observation data) necessary to define the trajectory.
2.2.1.1	REED	REED is designed as a simple special purpose read package to provide the capability for inputting modified station data and estimated satellite data.
2.2.1.1.1	SICRD	A special purpose routine designed to assure that the first data card is the station identification card.
2.2.1.2	INITAL	INITAL sets up the solution process by defining all of the required vectors in the computational coordinate frame of 1950.0 from the information given in the true equation of data frame.
2.2.1.3	POSVEL	This routine is designed to provide the user the option of reading either the satellites position and velocity vectors at the initial epoch or the equivalent set of orbital elements.
2.2.1.4	BLOCK	Data for the lunar and solar ephemerides and for the 1962 U. S. Standard atmosphere are stored in this routine. These data are loaded directly into memory at the time the program is loaded (BLOCK is not called).
2.2.1.5	DADUMP	A special purpose routine designed to DUMP the entire blank COMMON array when called.
2.3.1	TRAJ	TRAJ mechanizes the working portion of the program and serves as the means whereby the reference trajectory is computed and the tracking stations of the problem checked.

SECTION	ROUTINE	DEFINITION
2.3.1.1	CONIC	CONIC generates the position and velocity on an arbitrary conic section defined by $\vec{r}_o$ $\vec{v}_o$ as a function of time. It is utilized for the purpose of defining the Encke reference trajectory and the luni-solar ephemerides.
2.3.1.1.1	SEARCH	This routine is designed to function in conjunction with CONIC for the purpose of iteratively solving Kepler's equation.
2.3.1.1.2	TIME	This routine defines time on the conic section as a function of a position anomaly.
2.3.1.1.3	PARTL	PARTL is the partial derivative of the time variable with respect to the anomaly variable. This information is utilized in a Newton iteration by SEARCH.
2.3.1.2	MOTION	MOTION is the driver routine for evaluating the total acceleration experienced by a vehicle moving in space relative to the Encke reference trajectory in the computational coordinate frame (1950.0).
2.3.1.2.1	OBLN	OBLN computes the acceleration resulting from the first three zonal harmonics of the Earth's potential function.
2.3.1.2.2	DRAG	This routine defines the acceleration produced by the tenuous atmosphere on a spherical satellite.
2.3.1.2.2.1	ATMS	ATMS operates in conjunction with DRAG and is designed to compute the instantaneous estimate of the atmospheric density.
2.3.1.2.3	ENCKE	Since the largest contribution to the acceleration is included in the reference trajectory and since off nominal motion produces a modification to the gravity vector, ENCKE is required to define the change in the acceleration resulting from the change in the central force.

SECTION	ROUTINE	DEFINITION
2.3.1.2.4	PRESS	PRESS computes the acceleration resulting from solar pressure on an equivalent spherical satellite.
2.3.1.2.4.1	SPOWER	This function operates in conjunction with PRESS. Its purpose is to define the ratio of the solar power to the speed of light.
2.3.1.2.5	PERT	PERT defines the gravitional acceleration experienced by the vehicle due to the sun and moon.
2.3.1.2.5.1	FQ	PERT is formulated in a manner similar to ENCKE and thus it employs a series to evaluate the off nominal nature of the acceleration. FQ is that series.
2.3.1.2.6	EPHEM	The position vectors for the sun and moon relative to the earth which are required by both PRESS and PERT are computed by EPHEM from data stored in BLOCK.
2.3.1.3	INGRAT	INGRAT is a driver routine for producing the first and second integrals of the output from MOTION.
2.3.1.3.1	HSIZE	This routine determines the optimum stepsize for a Gauss-Jackson intregration of the equations of motion.
2.3.1.3.2	START	START is a fourth order Runge-Kutta integration routine utilized to establish the data necessary to produce the difference table utilized in the Gauss-Jackson process.
2.3.1.3.3	DIFTAB	This routine differences the acceleration vectors established by START and evaluates the first and second sums on the leading diagonal of the difference table.
2.3.1.3.4	INTEG	INTEG mechanizes the Gauss-Jackson integration formulas for a stepwise integration of the equations of motion. INTEG also steps the leading diagonal in the difference table.

SECTION	ROUTINE	DEFINITION
2.3.1.4	TRAK	TRAK is the driver program which checks the tracking stations of the problem and determines if any can observe the satellite at that time. TRAK transfers to FILTER if observation data are available at this time.
2.3.1.4.1	UNIT	UNIT computes the position vector for the tracking station and establishes the up, east, north unit vectors at the station.
2.3.1.4.2	EQINOX	This routine defines the coordinate transformation relating the observational coordinate frame (true equator of date) and the computational frame (1950.0). This transformation is the result of nutation and precession.
2.3.1.4.3	GHA	GHA defines the Greenwich Hour Angle as a function of universal time and the number of days since 1950.0.
2.4.1	FILTER	FILTER is the driver routine for a Kalman estimate of the correction to position and velocity (state vector) resulting from processing the observed data.
2.4.1.1	KALMAN	This routine computes the minimum variance estimate of the state vector and the covariance matrix for the estimation errors.
2.4.1.2	STMAT	The errors at two points on the perturbed trajectory are related by employing an averaging process and two estimates of the transition matrix obtained using conic representation (TRANS).
2.4.1.2.1	TRANS	TRANS provides STMAT the conic approximations of the transition matrices.

SECTION	ROUTINE	DEFINITION
2.4.1.2.2	INVAO	This routine works in conjunction with STMAT in the averaging process. It is designed to determine the analytic inverse of the output of TRANS.
2.4.1.3	MEASUR	MEASUR computes the matrix of partial derivatives of the observation vector with respect to the state vector.
2.4.1.4	ERROR	This routine computes the weighting matrix for the Kalman estimate. Data for Station errors and noise in the observations are employed.
2.4.1.5	UPSTAT	UPSTAT is designed to determine whether the state vector as computed by KALMAN is known to a degree which is satisfactory to allow the position and velocity to be corrected.
2.4.1.5.1	ELEMEN	Since the orbital elements of the osculating conic are of interest in the data reduction problem, ELEMEN is included to provide this information at those times when data is processed.
2.4.1.6	DATAPE	DATAPE defines the time of the next observation, the recording station, the type of data and the data itself. This information is recorded on magnetic tape by the preprocessor.
2.5	Mathematical Subroutines	
2.5.1	MATMPY	Matrix multiplication of two conformable matrices in double precision.
2.5.2	MTINV	Matrix inverse of a square matrix in double precision.

SECTION	ROUTINE	DEFINITION
2.5.2.1	CHOOSE	This routine functions in conjunction with MTINV to determine if the matrix is singular.
2.5.3	ADDMAT	Matrix addition
2.5.4	SUBMAT	Matrix subtraction
2.5.5	TRANSP	Matrix transposition
2.5.6	CROSS	Vector cross product
2.5.7	DOT	Vector scalar product
2.5.8	AMAG	Vector magnitude
2.6.1	SINH	Hyperbolic sine
2.6.2	COSH	Hyperbolic cosine
2.6.3	ARKTNS	Computes the single precision arc tangent and assigns the angle to the proper quadrant
2.6.4	DERAQ	The Kronecker delta

BLANK PAGE

1. INTRODUCTION

1.1 PURPOSE

FS4-305 is a FORTRAN IV IBM 7094 program which was written and checked using the standard North American Aviation monitor system (NAASYS - version 13). However, the program can also be operated on CDC equipment and other systems possessing FORTRAN capabilities. Consequently, to minimize possible system incompatibilities, care has been exercised to assure that only the basic features of the system are utilized. - This approach assures that the program will be operable on new generation machines (e.g., the IBM system 360).

The primary function of this program is to construct an accurate trajectory of a geocentric satellite based on a series of observations (e.g., range, range-rate, azimuth and elevation, etc.). This task is performed through a differential corrections process by generating a reference trajectory and minimizing the observed minus computed residuals in a minimum variance sense. (The operation of the filter is explained in general in section 1.2 and in detail in section 2.4.)

The trajectory required in the corrections process is generated by numerical integration of the accelerations relative to a reference conic (Encke's method). Accelerations resulting from the off-nominal nature of the motion, the Earth's oblateness (first 3 zonal harmonics), solar-lunar gravitation, atmospheric drag, and solar radiation pressure are all included. Tracking station data and the corresponding relative position data are generated simultaneously as a function of universal time for the purpose of defining the predicted values of the observations from each station.

The observation data utilized in this program are provided on a magnetic tape by the preliminary processor (see FS4-305A, SID 65-1203-2). These data have been operated on in two distinct fashions. First, the raw data from each tracking station were smoothed by filtering the observations over a restricted interval of time (approximately 20 seconds) to a parabolic curve and the midpoint of this interval was selected for processing in the differential corrections program. This operation assures that the effect of random scatter in the data can be reduced and assures reasonably efficient operation of the differential corrections process by reducing the amount of raw data to be considered. Secondly, the data from the various sites were identified as to the station which recorded them and the type of information gathered; then the complete array was arranged in a chronological fashion to facilitate filtering of the data in a recursive mode.

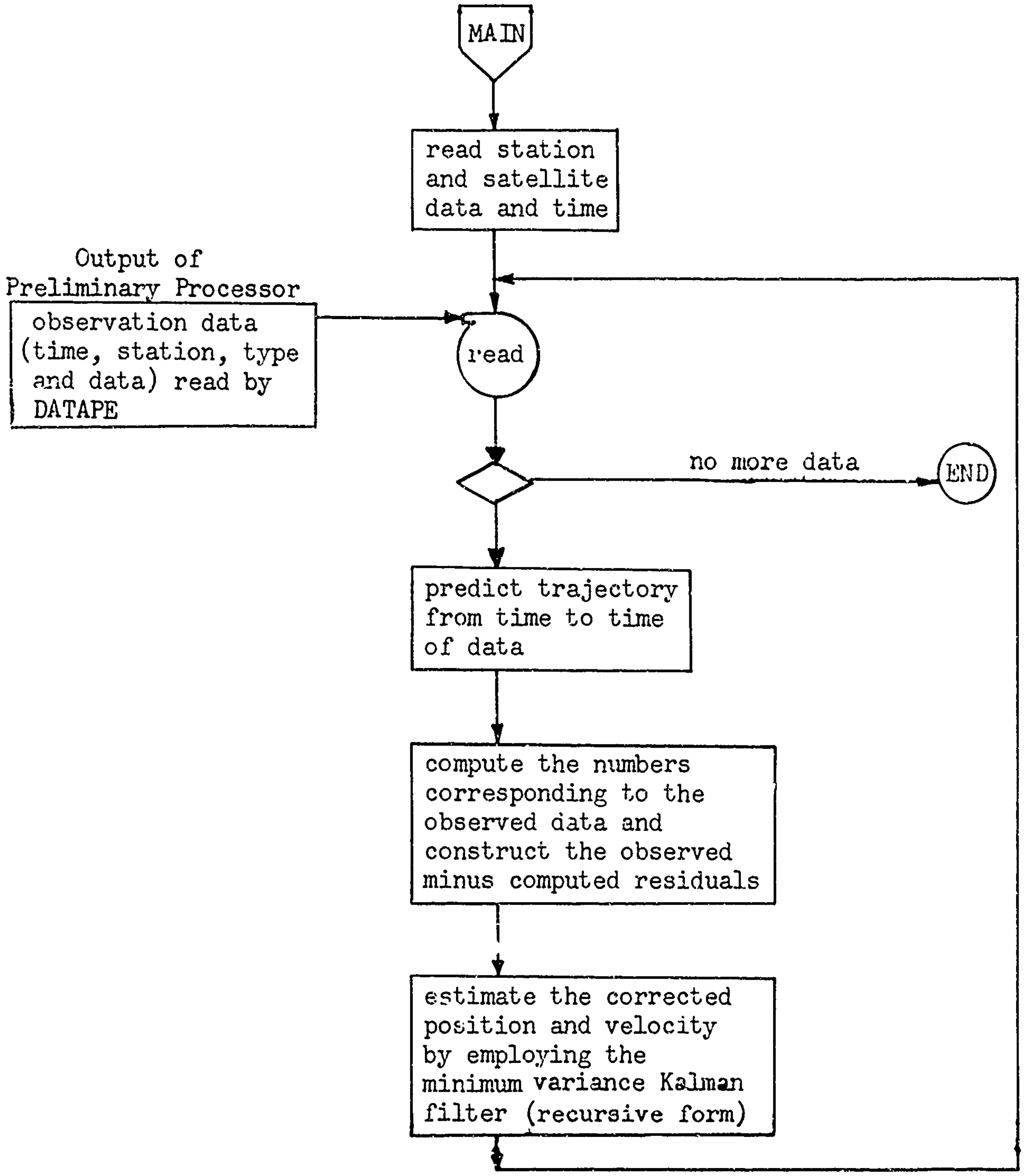
The interface of the preliminary processor and this program is provided by subroutine DATAPE (data tape). This routine reads the tape and identifies the data for use in differential corrections solution. Thus, complete consistency is assured.

1.2 Program Concept

The differential corrections orbit computation program is designed around the Kalman data filter developed and discussed in Subroutine KALMAN. Operation requires that input data be read to identify the stations which recorded the observed data, and estimates of the position and velocity (or orbital elements) of the vehicle at an arbitrary epoch. At this time, the first data point (time, type of data, recording station, and the observation vector) is read into memory and the trajectory from the initial epoch to the time of the observation is defined. When this is done, the position and velocity relative to the observing station are computed and an error signal is generated based on this data and the actual information recorded.

The filtering of this information or prediction of a corrected position and velocity is accomplished recursively (i.e., one set of observations at a time) by employing a minimum variance formulation (see KALMAN). This filter has been shown to approximate the true non-linear process very well. Further, the filter provides for weighting the data in a general manner not obtained with a simple least squares solution. Its inclusion is thus felt to constitute near optimum design of the differential corrections orbit computation program.

After predicting the best estimate of the position and velocity at the first data point, a second data point is read and the process is repeated. This operation is demonstrated in the following sketch.



An appreciation of the general nature of this program is now available. The immediate problem is, thus, to add detail to the flow diagram and establish a rationale which will assure efficient and accurate operation. This detail can best be included by dividing the program into logical blocks (driver subroutines) which accomplish the desired functions, and by discussing each block in terms of the intended purpose and formulation. Indeed, this division has been performed and is outlined fully in the discussion of the three principal drivers of the program, MAIN, TRAJ and FILTER. Therefore, attention will turn to the first, MAIN.

2.0 PROGRAM DISCUSSION

The formulation, rationale and computational logic for the routines designed to perform the differential correction of geocentric satellite trajectories is presented in the sections which follow. These discussions attempt to document each routine in a complete manner so that subsequent modification to any function presently being performed can be facilitated.

Careful review of these separate discussions is essential to establish the complete computational logic for the program and a working knowledge of its structure. However, if interest is centered in the area of program utilization, attention should be directed primarily into the structure of the sample problem and the associated input formats. (In the latter case, no detailed information of the program logic is essential.)

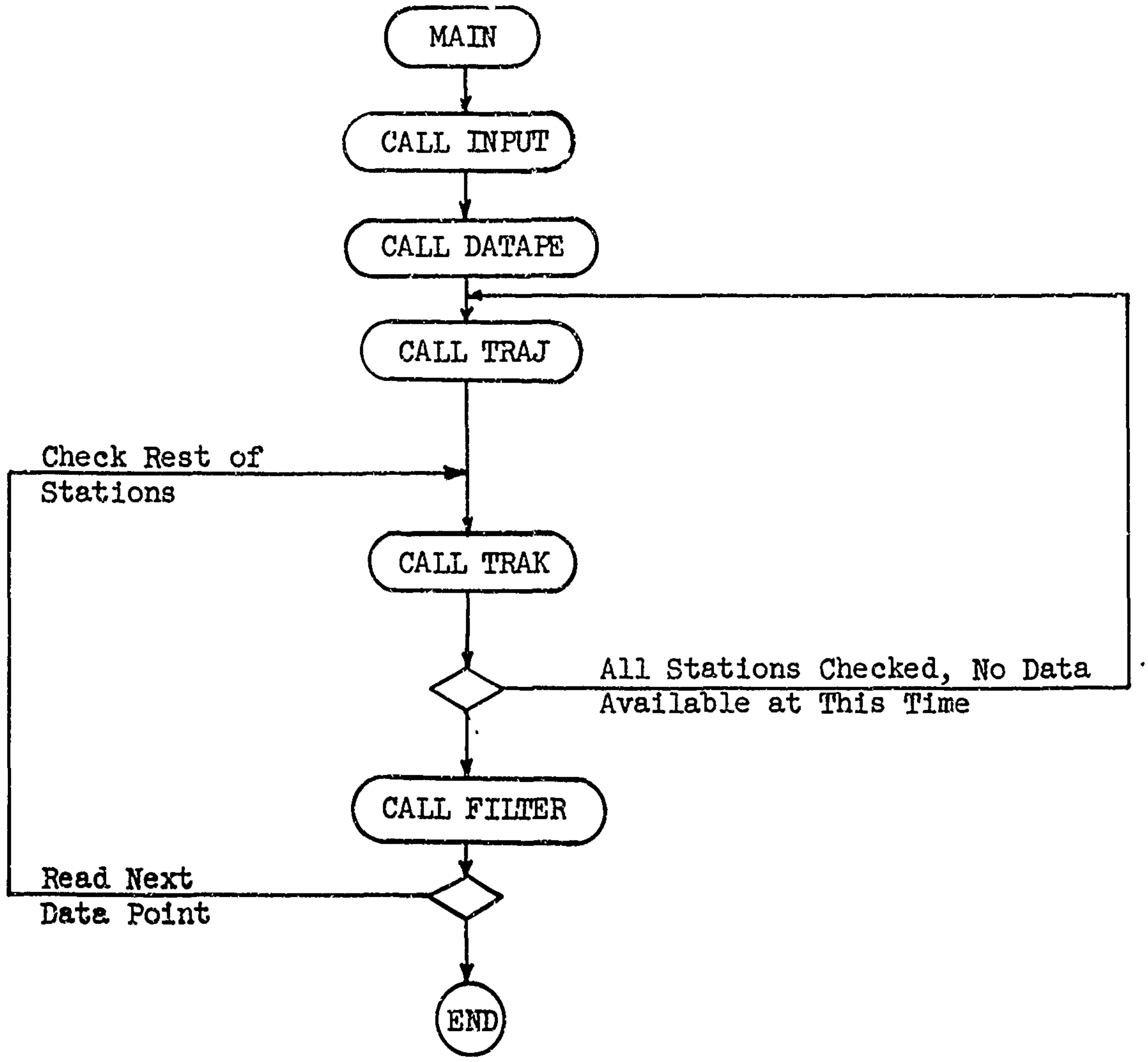
The mechanism for communicating between the routines of the program is the COMMON region DATA. A map of this region and the definition of the variables assigned is presented in Appendix 1. This information should be reviewed in conjunction with <u>any</u> attempt to revise the program logic.

2.1 MAIN Program

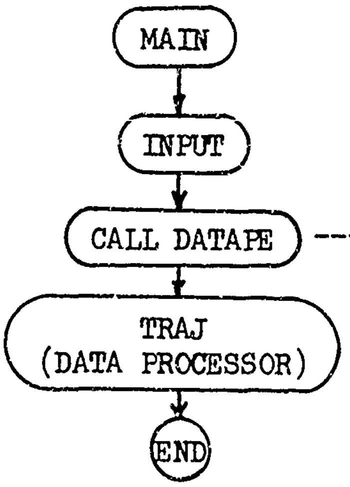
Purpose:	To mechanize the subroutines and lower order driver routines designed to effect the differential correction of a satellite's position and velocity vectors based on data recorded on a magnetic tape by the preliminary processor routine, PROCES.
Deck Name:	MAIN
Calling Sequence:	None
Input/Output:	None
Subroutines Required:	INPUT (loads satellite and station data) DATAPE (loads observation data) TRAJ (generates trajectory)
Functions Required:	None
Approximate Deck Length:	63 (octal)

Description and Program Logic:

MAIN is the driver routine which mechanizes the complete program for the purpose of processing observed data for a single (i.e., one at a time) identifiable satellite and differentially correcting the estimate of the position and velocity vectors describing its motion. In order to accomplish this task efficiently and facilitate the development and checkout process, the program has been constructed in a series of major blocks (modules), each of which is itself subdivided into a number of subroutines. These modules serve to provide the input data for the program, to read into memory the first observed data point from the data tape, to generate an accurate trajectory which can be utilized in the computation of the observed minus computed residuals, to check the various tracking stations employed in the problem for the relative position and velocity information, and to process the observed data (recorded on a magnetic tape provided by PROCES) and differentially correct the elements of the trajectory. In its simplest form, then, the program can be viewed as the link between these modules, i.e.,



However, due to the fact that data required as input to TRAK and FILTER are available in TRAJ and TRAK respectively and are not utilized in the MAIN Program for any other purpose, this logic has been modified slightly. The modification consists of placing the call statement for TRAK (and FILTER) in TRAJ (and TRAK) and constructing a transfer within TRAJ which will assure that control is retained by TRAJ until all data have been processed. In this sense, then, TRAJ is the gross data processor and the flow diagram reduces to:



This routine is documented with the Filter Group since successive calls will always come following the reduction of a given observation.

A careful review of these modules is essential to a thorough appreciation of the functioning of the program. For this reason, complete documentation of all routines employed will be presented in the following sections of this document. The task of reviewing this information, however, has been simplified by grouping the subroutines when their function relates them to a larger problem to assist in the comprehension.

```
      FS305                                                                              11/23/85
          MAIN        -  EFN   SOURCE STATEMENT  -  IFN(S)  -

C                                                                         MAIN0020
C     MAIN  IS THE DRIVER ROUTINE WHICH MECHANIZES THE COMPLETE PROGRAM.MAIN0030
C              THIS ROUTINE SERVES THE FOLLOWING FUNCTIONS *              MAIN004C
C                 1) MAIN CALLS THE INPUT PACKAGE                         MAIN0050
C                 2) MAIN CALLS DATAPE AND LOADS DATA FROM PROCES         MAIN0060
C                 3) MAIN CALLS TRAJECTORY AND RELINQUISHES CONTROL .     MAIN0070
C     NOTICE THAT ALL PARAMETERS ARE GIVEN INITIAL VALUES PRIOR           MAIN0080
C              TO CALLING TRAJEC EXCEPT THE STEP SIZE FOR THE NUMERICAL MAIN0090
C              INTEGRATION . THIS QUANTITY IS GUESSED IN HSIZE THE        MAIN0100
C              FIRST PASS THROUGH INGRAT .                                MAIN0110
C     ONCE EXITED,MAIN ISNT REENTERED UNTIL DATAPE(WHICH IS CALLED IN     MAIN0120
C              FILTER) INDICATES THAT THERE IS NO MORE DATA .AT THIS      MAIN0130
C              TIME,DATAPE CALLS THE $IBFTC NAME MAIN SO THAT ANOTHER     MAIN0140
C              PROBLEM (I.E.,ANOTHER TRAJECTORY ) CAN BE ESTIMATED .      MAIN0150
C     EXIT FROM MAIN IS ACHIEVED WHEN AN END OF FILE RATHER THAN THE      MAIN0160
C              ADDITIONAL PROBELM(WHICH IS EXPECTED) IS READ. THIS MODE MAIN0170
C              OF OPERATION MAY NOT BE CONSISTENT WITH ALL SYSTEMS .      MAIN0180
C                                                                         MAIN0190
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * MAIN0200
C                                                                         MAIN0210
      DIMENSION  CON( 15)  ,SAT( 20)  ,SDA(250)  ,STT(105)  ,WRK(135)     MAIN0220
C                                                                         MAIN0230
      COMMON DATA(1)                                                      MAIN0240
C                                                                         MAIN0250
      EQUIVALENCE          (DATA(  1),CON), (DATA( 16),SAT.               MAIN0260
     2                    ,(DATA( 36),SDA), (DATA(286),STT)               MAIN0270
     3                    ,(DATA(391),WRK)                                MAIN0280
C                                                                         MAIN0290
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * MAIN0300
C                                                                         MAIN0310
      DO 5  I=1,525                                                       MAIN0312
      DATA(I)=0.0                                                         MAIN0314
    5 CONTINUE                                                            MAIN0316
      CALL  INPUT                                                         MAIN0320      9
      KOUNT = 2                                                           MAIN0330
      CALL DATAPE( KOUNT )                                                MAIN0340     12
```

```
      MAIN        -  EFN   SOURCE STATEMENT  -  IFN(S)

      CALL TRAJ                                                   MAIN0350
C                                                                 MAIN0360
C     IF CONTROL RETURNS TO MAIN AT THIS POINT THERE IS AN ERROR IN   MAIN0370
C           THE PROCEDURE EMPLOYED IN DATAPE                      MAIN0380     14
      WRITE(6,10)                                                 MAIN0390     15
   10 FORMAT(37H1 CONTROL RETURNED TO MAIN IMPROPERLY  )          MAIN0400
      CALL DUMP                                                   MAIN0410     16
      RETURN                                                      MAIN0420
      END                                                         MAIN0430
```

MAIN

STØRAGE MAP

MAIN PRØGRAM

COMMØN VARIABLES

COMMØN BLØCK // ØRIGIN 0CC01 LENGTH 01015

SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE
DATA	C0000	R	DATA	00000	R	CØN	00000	R
SAT	00017	R	SDA	00043	R	STT	00435	R
WRK	00606	R						

UNDIMENSIØNED PRØGRAM VARIABLES

SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE
KØUNT	C1016	I						

ENTRY PØINTS

...... SECTIØN 4

SUBRØUTINES CALLED

INPUT	SECTIØN	5	DATAPE	SECTIØN	6	TRAJ	SECTIØN	7
.FWRD.	SECTIØN	8	DUMP	SECTIØN	9	.UN06.	SECTIØN	1C
.FFIL.	SECTIØN	11	.FCNV.	SECTIØN	12	SYSLØC	SECTIØN	13

EFN IFN CØRRESPØNDENCE

EFN	IFN	LØCATIØN	EFN	IFN	LØCATIØN	EFN	IFN	LØCATIØN
5	6A	C1040	10	FØRMAT	01026			

DECK LENGTH IN ØCTAL IS C0071.

2.2 The INPUT Group

This group of routines is designed for the purpose of providing the communications link between the users of the differential corrections program and the program itself. The group contains a routine which provides nominal information to the program (INPUT), a routine which reads specific information regarding the satellite (REED), a routine which loads ephemeris and atmosphere information (BLOCK), a routine which effects conversion from orbital elements to position and velocity if this form of the data is provided (POSVEL), and a routine to set up the solution process by initializing various data cells in memory (INITAL).

The group also includes in a sense the routine (DATAPE) which reads the data tape prepared by the preliminary processor. However, since this routine is a more integral part of the logic associated with the filtering of the data, it is documented in the FILTER group.

2.2.1 Subroutine INPUT

Purpose: INPUT provides all of the data (except the observation data) necessary to effect the differential corrections solution of a given trajectory. Data for the physical characteristics of the Earth, Sun, Moon, and vehicle, station location and error data, position and velocity information for the satellite, and an estimate of the covariance matrix for errors in the state vector at the initial epoch are provided.

Deck Name: INPUT

Calling Sequence: CALL INPUT

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
O	CON	-	15	DATA (1)	A subarray of common containing constants for the program
O	RE RPOL COEFJ COEFH COEFD GMERTH OMEGA	R_e R_p J H D μ ω	1 1 1 1 1 1 1	CON (1) CON (2) CON (3) CON (4) CON (5) CON (6) CON (7)	Equatorial and Polar radii (Km) for the reference ellipsoid, the first 3 coefficients of the potential function (Jefferies notation), the gravitational constant (Km^3/sec^2), and the spin rate for the Earth.
O	GMMOON GMSUN	μ_m μ_s	1 1	CON (8) CON (9)	Gravitational constants for the moon and sun (Km^3/sec^2)
O	AU	AU	1	CON (10)	The astronomical Unit (Km)
O	CONDAY	$\frac{SEC}{DAY}$	1	CON (11)	Conversion from days to seconds (86400.)
	CONV 2	$\frac{RAD}{DEG}$	1	CON (12)	Conversion from degrees to radians
O	NIN NOUT	-	1 1	CON (13) CON (14)	Input and output tape numbers

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	SAT	-	20	DATA (16)	A subarray of common containing data pertaining to the satellite at the initial epoch.
I/O	SMASS AREA CD REFLEC	m A C_D R	1 1 1 1	SAT (1) SAT (2) SAT (3) SAT (4)	Satellite mass (K_g), reference area (m^2), drag coefficient, and surface reflectivity.
I/O	RVEC	$\vec{r}$ (or a,e,i) $\vec{v}$ (or Ω,ω,θ_o)	3 3	SAT (5) SAT (8)	Position and velocity vectors (or orbital elements) in the true equator of date frame at the initial epoch (Km-sec Units).
I/O	TW (DJ) TF (DJF)	t_o	1 1	SAT (11) SAT (12)	The whole and fractional part of the day corresponding to $\vec{r}$, $\vec{v}$ relative to January 0 1950. (1950.0) J.D. 2433282.423.
I/O	WINDEX CHECK GO NO CODUMP	- - - -	1 1 1 1	SAT (13) SAT (14) SAT (15) SAT (16)	Indices utilized to define the format of RVEC and VVEC (cartesian vectors if 1. or orbital elements if 2.), whether or not each station is to be checked at each integration step (yes if 1. or no if 2.), whether or not trajectory data is to be prepared for each N^{th} integration step (yes if N. or no if 0.), and whether COMMON is printed (no if zero).
O	KCHECK NOGO	- -	1 1	SAT (17) SAT (18)	Fixed point equivalents of SAT (14) and SAT (15).
I	SDA	-	250	DATA (36)	A subarray of common containing station data.
I	N (I)	n_i	10	-	Station identification indices - used to select those stations to be employed.

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
O	STATN	I λ H Name	40	SDA (1)	Station array (1 to 10) containing position information for each of the stations and a 6 character identification.
O	HORCOR	E	10	SDA (41)	Horizon corrections in elevation for each of the 10 stations.
O	STERR	2	30	SDA (51)	Station error array (1 to 10 stations) containing variance information for latitude, longitude and altitude (_, _, Km^2)
O	SNOISE	2	40	SDA (141)	Station noise array, (1 to 10 stations) containing variance information for measurement for errors in range, range in rate, azimuth and elevation, (Km^2, Km^2/sec^2, _, _,)
O	Number	-	1	SDA (241)	The total number of stations employed in the tracking program for this given simulation
I	STT	-	105	DATA (286)	A subarray of common containing statistical information (and related data) for the satellite
I/O	PP	P	6 X 6	STT (70)	Covariance matrix for the errors in $\bar{r}$ and $\bar{v}$ at t_o (metric units) resulting from smoothing raw data as done in the preliminary processor.

Subroutines Required:	REED	(Relative Read package)
	POSVEL	(Converts elements to $\vec{r}_0$, $\vec{v}_0$)
	INITAL	(initializes vectors for problem)
Functions Required:	None	
Approximate Deck Length:	1452 (octal)	

Description:

INPUT provides all of the constants, the data for a fixed network of ten tracking stations, and the logic for selecting (and modifying) from the fixed array of stations those to be employed on the analysis. In addition, INPUT serves as the driver for a read routine (REED) which is constructed in such a manner that any of the constants previously loaded into common can be changed at the time the physical data and estimated orbital data for the satellite being tracked are loaded. Upon return from REED a test is made to determine whether the 6 components of position and velocity or 6 constants of integration in the form of the orbital elements (a, e, i, Ω, ω, θ_0) were provided. (In the latter case the corresponding radius and velocity vectors are computed). INPUT then calls subroutine INITAL which initializes the cells in the common array containing vectors used in the integration of the "best" trajectory and in the processing of the data. As a final step, all of the input data are printed for reference.

The first observation which should be made of INPUT is that variance data are given for the station errors (i.e., variance in latitude, longitude, and altitude) and for the noise errors (i.e., variance in range, range-rate, azimuth, and elevation) under the assumption that the errors are uncorrelated. If calibration of the stations employed in this simulation should subsequently prove that the present formulation is inadequate or could be improved, the complete arrays should be provided. The incorporation of these additional data requires only that the dimension of the SN, SE, SNOISE, and STERR arrays be altered and that the FILTER package be modified accordingly to accept the new data. Allowances for this modification have been made in the design of the common array in that extra locations have been provided.

A map of the COMMON array (DATA) utilized by this routine is presented as Appendix 1 to this report. This map should be utilized to facilitate communication with the program and to aid in analysis of the communication between routines.

The second observation which should be made is that, since the data being processed have been smoothed previously to reduce the number of raw data points and to remove the random scatter in the raw data, the variances utilized in the program should correspond to the smoothed data points rather than the raw data. Thus, to prepare these data if raw data variances are known, it is necessary to develop the covariance matrix for the difference in the true and smoothed values of the observation vector.

$$\Delta\vec{\Psi} = \vec{\Psi}_{TRUE} - \vec{\Psi}_{SMOOTH}$$

and note that the smoothed values obey the equation

$$Y_{SMOOTH} = \begin{bmatrix} 1 & X_1 & X_1^2 \\ 1 & X_2 & X_2^2 \\ & \vdots & \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

$$\equiv \quad M \; \vec{c}$$

where: $X_1 = t_1 - t_{midpoint}$

a,b,c = coefficients of the parabola which best fits the data.

AL

Further, the vector of constants ($\vec{c}$) which is determined in an unweighted least squares fit of the parabola to the data can be represented as

$$\vec{c} = \phi \vec{\Psi}_{observed}$$

(see the preliminary processor documentation of SMOOTH). Now, if the observed vector is represented as

$$\vec{\Psi}_{observed} = \vec{\Psi}_{true} + \vec{N},$$

substitution back into the equation for $\Delta\vec{\Psi}$ yields

$$\Delta\vec{\Psi} = (I - M\phi)\,\vec{Y}_{TRUE} - M\phi\vec{N}$$

$$\equiv A\vec{\Psi}_{TRUE} - K\vec{N} \quad .$$

Thus, the desired covariance matrix can be found by taking the expected value of the product $\Delta\vec{\Psi}\,\Delta\vec{\Psi}^T$.

$$E\left(\Delta\vec{\Psi}\;\Delta\vec{\Psi}^T\right) = E\left\{A\vec{\Psi}_T\vec{\Psi}_T^T A^T - A\vec{\Psi}_T\vec{N}^T K^T - K\vec{N}\vec{\Psi}_T^T A^T + K\vec{N}\vec{N}^T K\right\}$$

But: $E\ (\vec{\varphi}_T\ \vec{\varphi}_T^{\ T}) \equiv 0$

$E\ (\vec{\varphi}_T\ \vec{N}^T) \equiv 0$

$E\ (\vec{N}\ \vec{\varphi}_T^{\ T}) \equiv 0$

$E\ (\vec{N}\ \vec{N}^T) \equiv V$

Leaving

$$E\ (\Delta\vec{\varphi}\ \Delta\vec{\varphi}^T) = K\ V\ K^T$$

which is generally not diagonal even for a diagonal V, but which (for the types of problems attempted during checkout) can be approximated with a diagonal matrix.

The operation of INPUT is controlled by data provided from REED and on the station identification card (SIC) which must precede all of the card input. This is accomplished as follows: the SIC (a card of 10 fixed point numbers of FORMAT 10 I 4) is read and those stations for which an index other than zero is recorded are assumed to be utilized in the data reduction problem (The numerical sequence of the utilized stations must be provided as input to the preliminary processor so that the observed data will be credited to the proper station). Now, INPUT selects data from arrays of numbers provided for these stations and loads these data into common for use in subsequent operations (it is thus important to note that the remaining station data are discarded and will not be available to the program at later time) subject to modification from REED if any or all of the data from one or more stations is to be changed.

Having selected the nominal station data, INPUT calls REED for the purpose of obtaining the required satellite data and varifying the nominal station arrays. REED provides data to the program by assigning any data (previous to the first blank 12 digit field) to the common location provided in the first 12 digit field (right justified) of the card. When REED recognizes the fixed point number 999 in the first field of a card it assumes all numerical data have been provided and begins to look for new station names (one to a card) until the location 9999 is read. At this point, all data are assumed to be available and control is returned to INPUT. It is important to note that with the exception of the grouping of the data cards before 999 and the name cards before 9999, no sequencing of the data cards for subroutine REED is required.

The data expected by INPUT from REED is as follows:

Data (16)	=	satellite mass (kg)
Data (17)	=	cross section area (m^2)
Data (18)	=	drag coefficient
Data (19)	=	surface reflectivity
Data (20)	=	radius vector (X, Y, Z) (Km) or the orbital elements a, e, i in the true equator of date frame (Km, - , DEG)
Data (23)	=	velocity vector (X, Y, Z) (Km/sec) or the orbital elements ω, Ω, θ (true anomaly) (DEG, DEG, DEG)
Data (26)	=	whole number of days since 1950.0 to beginning of present date
Data (27)	=	fractional part of present day in U.T. expressed in days

Data (28) = WINDEX - an index utilized to define the option for Data (104) - (109). WINDEX = 1. if these data are position and velocity vectors; and 2. if they are orbital elements

Data (29) = CHECK - an index employed in TRAK to bypass checking the tracking stations unless tracking data are available at the given instant. CHECK = 1. no bypass; = 2. for efficiency

Data (30) = GONO - an index employed in TRAK to bypass checking the tracking stations unless the particular station involved recorded the data. GONO = 1. no bypass; = 2., bypass

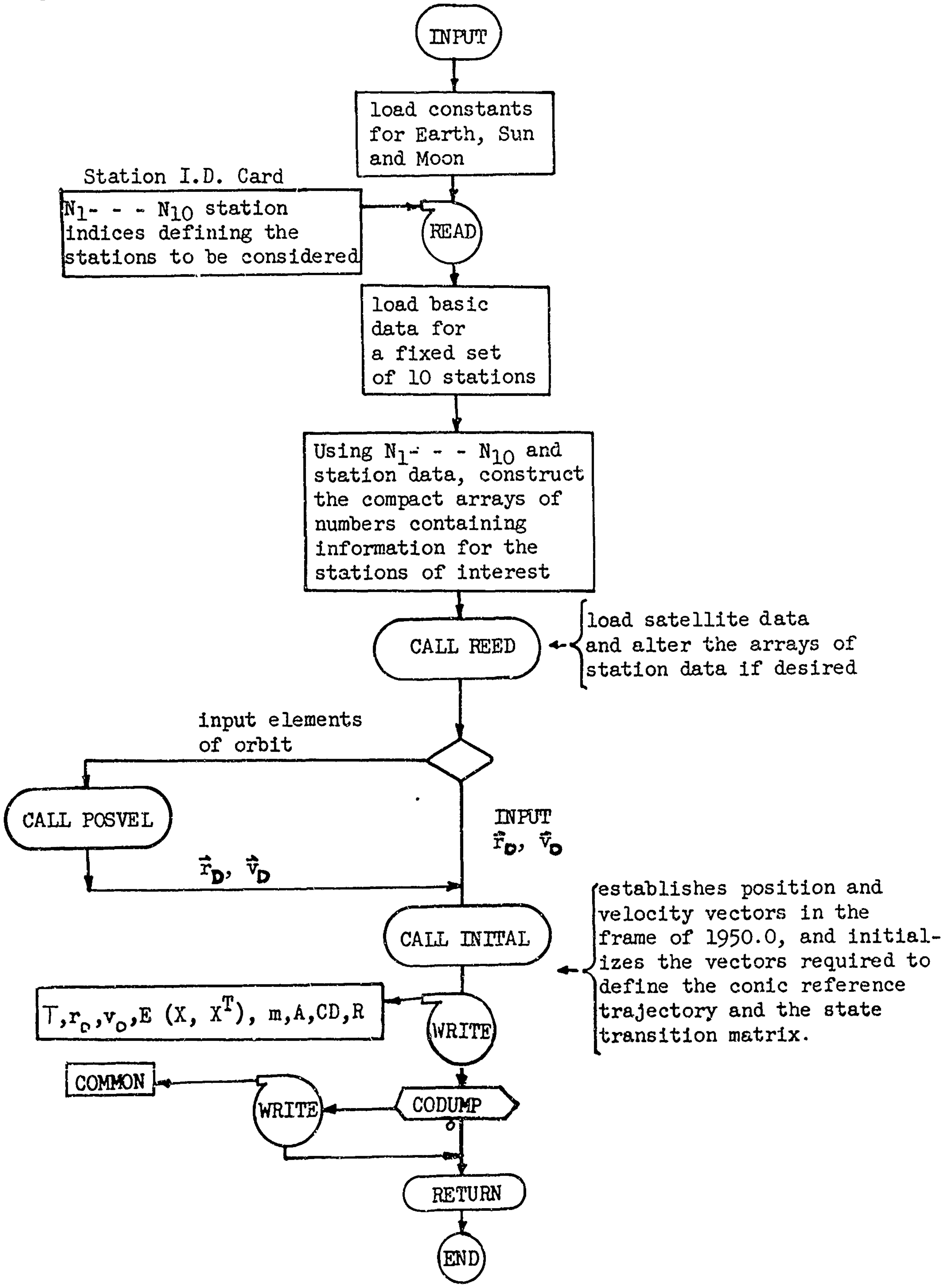
Data (31) = CODUMP - an index used to assist checkout and as a permanent record of program constants, etc. If CODUMP = nonzero, the entire DATA array is printed following initialization.

Data (356) = covariance matrix for the errors in the initial position and velocity vectors in the coordinate system of $\vec{r}_o$ and $\vec{v}_o$ and in units of Km^2 and $(KM/sec)^2$

These data can be augmented with station data if desired. This step is accomplished by identifying the number of the station to be changed (the number of stations on the SIC preceding the station to be changed) and loading data into the following locations.

Data (36 + 4N)	=	Latitude in radians
Data (37 + 4N)	=	Longitude in radians
Data (38 + 4N)	=	Altitude relative to the reference ellipsoid (KM)
Data (39 + 4N)	=	Station name (6 characters)
Data (77 + N)	=	Horizon correction (rad)
Data (87 + 3N)	=	Variance in latitude $(rad)^2$
Data (88 + 3N)	=	Variance in longitude $(rad)^2$
Data (89 + 3N)	=	Variance in altitude $(KM)^2$
Data (177 + 4N)	=	Variance in range $(KM)^2$
Data (178 + 4N)	=	Variance in range-rate $(KM/sec)^2$
Data (179 + 4N)	=	Variance in azimuth $(rad)^2$
Data (180 + 4N)	=	Variance in elevation $(rad)^2$

Computational Logic:



```
      SUBROUTINE INPUT                                                  IPUT0020
C                                                                       IPUT0030
C     THIS ROUTINE CONTAINS ALL OF THE REQUIRED INPUT DATA AND READ     IPUT0040
C             INFORMATION * IT SHOULD BE REVIEWED CAREFULLY TO ASSURE   IPUT0050
C             COMPATIBILITY WITH DESIRED TRAJECTORY AND STATIONS        IPUT0060
C                                                                       IPUT0070
C                                                                       IPUT0080
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * IPUT0090
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * IPUT0100
C                                                                       IPUT0110
C                                                                       IPUT0120
C     THE FOLLOWING SECTION OF THIS ROUTINE LOADS PHYSICAL CONSTANTS    IPUT0130
C             INTO THE PROGRAM IN MKS UNITS .  INCLUDED                 IPUT0140
C             IN THIS COLLECTION ARE CONVERSION DATA , CONSTANTS FOR    IPUT0150
C             THE EARTH,MOON,SUN,INPUT AND OUTPUT TAPE NUMBERS.         IPUT0160
C                                                                       IPUT0170
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * IPUT0180
C                                                                       IPUT0190
      DIMENSION  CON(1), SAT(1), SDA(1), WRK(1), STT(1)                 IPUT0200
C                                                                       IPUT0210
      COMMON DATA(1)                                                    IPUT0220
C                                                                       IPUT0230
      EQUIVALENCE          (DATA(  1),CON), (DATA( 16),SAT)             IPUT0240
     2                    ,(DATA( 36),SDA), (DATA(286),STT)             IPUT0250
     3                    ,(DATA(391),WRK)                              IPUT0260
C                                                                       IPUT0270
      DIMENSION      RVEC(3)     , VVEC(3)     , STATN(40)  , HORCOR(10) IPUT0280
     2, STERR(90)  , SNOISE(100), PP(6,6)      , P(10,4)    , SN(10,4)   IPUT0290
     3, SE(10,3)   , HC(10)      , N(10)                                 IPUT0300
C                                                                       IPUT0310
      EQUIVALENCE          (CON(  1),RE     ), (CON(  2),RPOL   )       IPUT0320
     2  ,(CON(  3),COEFJ ), (CON(  4),COEFH  ), (CON(  5),COEFD  )       IPUT0330
     3  ,(CON(  6),GMERTH), (CON(  7),OMEGA  ), (CON(  8),GMMOON)       IPUT0340
     4  ,(CON(  9),GMSUN ), (CON( 10),AU     ), (CON( 11),CONDAY)       IPUT0350
     5  ,(CON( 12),CONV2 ), (CON( 13),NIN    ), (CON( 14),NOUT   )       IPUT0360
C                                                                       IPUT0370
```

```
      EQUIVALENCE            (SAT(  1),SMASS ), (SAT(  2),AREA  )          IPUT0380
     2  ,(SAT(  3),CD     ), (SAT(  4),REFLEK), (SAT(  5),RVEC  )          IPUT0390
     3  ,(SAT(  8),VVEC   ), (SAT( 11),DJ    ), (SAT( 12),DJF   )          IPUT0400
     4  ,(SAT( 13),WINDEX), (SAT( 14),CHECK ), (SAT( 15),GONO  )          IPUT0410
     5  ,(SAT( 16),CODUMP), (SAT( 17),KCHECK), (SAT( 18),NOGO  )          IPUT0420
C                                                                          IPUT0430
      EQUIVALENCE            (SDA(  1),STATN ), (SDA( 41),HORCOR)          IPUT0440
     2  ,(SDA(51),STERR  ), (SDA(141),SNOISE), (SDA(241),NUMBER)          IPUT0450
C                                                                          IPUT0460
      EQUIVALENCE            (STT( 70),PP     )                            IPUT0470
C                                                                          IPUT0480
C                                                                          IPUT0490
      DATA V1,V2,V3,V4,V5,V6,V7,V8,V9,V10 /  5HFLOYD,5HANTIG,5HASCEN      IPUT0500
     1,5HJOHAN,6HJODREL,6HWOMERA,6HAMR-GE,6HHAWAII,6HBRMUDA,6HMSTONE/    IPUT0510
C                                                                          IPUT0520
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  IPUT0530
C                                                                          IPUT0540
C     DAYS TO SEC                                                          IPUT0550
      CONDAY = 86400.                                                      IPUT0560
C     DEGREES TO RADIANS                                                   IPUT0570
      CONV2  = 3.14159265/ 180.                                            IPUT0580
C     INPUT AND OUTPUT TAPE NUMBERS                                        IPUT0590
      NIN = 5                                                              IPUT0600
      NOUT= 6                                                              IPUT0610
C     CONSTANTS FOR THE EARTH                                              IPUT0620
      RPCL = 6356.777                                                      IPUT0630
      RE   = 6378.163                                                      IPUT0640
      GMERTH=398603.2                                                      IPUT0650
      COEFJ= 3.*(1082.30E-6)/ 2.                                           IPUT0660
      COEFH= -5.75E-6                                                      IPUT0670
      COEFD= +6.75  E-6                                                    IPUT0680
      OMEGA= 6.28318593/ 86164.091                                         IPUT0690
C     CONSTANTS FOR THE SUN AND MOON                                       IPUT0700
      AU = 149.53E6                                                        IPUT0710
      GMSUN = 1.3271545E11                                                 IPUT0720
      GMMOON= 4900.7588                                                    IPUT0730
C                                                                          IPUT0740
```

```
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * IPUT0750
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * IPUT0760
C                                                                      IPUT0770
C     THE FOLLOWING SECTION OF ** INPUT ** IS DESIGNED TO PROVIDE THE   IPUT0780
C              REQUIRED TRACKING SYSTEM DATA . THE ROUTINE HAS DATA     IPUT0790
C              FOR 10 STATIONS BUILT INTO THE ROUTINE THOUGH ANY        IPUT0800
C              GROUP OF THESE 10 STATIONS CAN BE CONSIDERED SEPARATELY. IPUT0810
C              THE ROUTINE ALSO PROVIDES FOR THE REPLACEMENT OF ANY OF  IPUT0820
C              THE 10 STATIONS THROUGH THE MEDIUM OF THE READ STATEMENT IPUT0830
C              AND THE SELECTIVE REPLACEMENT OF PORTIONS OF THE DATA    IPUT0840
C              THE SAME MANNER .                                        IPUT0850
C     THE ROUTINE IS SETUP TO UTILIZE THE STATION DATA DEPENDING ON     IPUT0860
C              A SERIES OF FIXED POINT NUMBERS WHICH MUST BE READ INTO  IPUT0870
C              THE PROGRAM.  THESE NUMBERS ( N(I), I=1,10 ) FORCE THE   IPUT0880
C              PROGRAM TO INCLUDE A STATION IF  N(I) IS NOT EQUAL TO    IPUT0890
C              ZERO( E.G. , N(I) = THE NUMBER OF THE STATION  )         IPUT0900
C     UPCN COMPLETION OF THE STATION SELECTION TASK SUBROUTINE READ IS  IPUT0910
C              CALLED FOR THE PURPOSE OF COMPLETING THE REQUIRED INPUT  IPUT0920
C              DATA TABLE. DURING THIS OPERATION THOSE STATIONS TO BE   IPUT0930
C              CHANGED WILL BE ALTERED AS FOLLOWS . THE LOCATION IN THE IPUT0940
C              DATA ARRAY IS LOCATED AND PLACED IN THE FIRST 12 DIGIT   IPUT0950
C              FIELD ON THE CARD ,THEN THE NEW NUMBER IS PLACED IN THE  IPUT0960
C              SECOND FIELD . SUCCEDING NUMBERS CAN BE CHANGED ON THE   IPUT0970
C              SAME CARD. BUT, IF THE LOCATION SEQUENCE IS BROKEN A     IPUT0980
C              NEW CARD MUST BE STARTED. SIMILARLY , IF THE             IPUT0990
C              STATN ARRAY IS MODIFIED,ALLOWANCE MUST BE MADE FOR THE   IPUT1000
C              FACT THAT THE FOURTH ELEMENT IN THE ARRAY IS NOT A       IPUT1010
C              NUMBER BUT IS RATHER A NAME ( THUS, THESE NAME DATA MUST IPUT1020
C              BE READ INTO MEMORY IN A SEPARATE MODE --AFTER ALL       IPUT1030
C              NUMERICAL DATA HAVE BEEN COMPLETED ) . THIS ALLOWANCE    IPUT1040
C              WAS MADE BY TESTING THE LOCATION AND TRANSFERING FROM    IPUT1050
C              THE DATA MODE TO THE NAME MODE WHEN 999 IS ENCOUNTERED.  IPUT1060
C              REED THEN READS NAMES (ONE LOCATION AND ONE NAME TO THE  IPUT1070
C              CARD)UNTIL 9999 IS READ. AT THIS POINT INPUT IS STOPPED. IPUT1080
C     IT IS IMPORTANT TO NOTE THAT THE NUMBER OF STATIONS IS NOT READ   IPUT1090
C              THUS IT IS NECESSARY TO HAVE THE PROPER                  IPUT1100
C              NUMBER OF NONZERO N(I) TO ASSURE PROPER PROGRAM FUNCTION.IPUT1110
```

```
C                                                                     IPUT1120
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * IPUT1130
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * IPUT1180
C                                                                     IPUT1190
C     THE DATA TABULATED ON THE FOLLOWING CARDS CONSTITUTE ALL OF THE  IPUT1200
C             STATION DATA REQUIRED FOR THIS PROBLEM . FOR CONVENIENCE,IPUT1210
C             THEY HAVE BEEN GROUPED BY STATION TO FACILITATE         IPUT1220
C             CHANGING DATA AT A LATER DATE IF DESIRED. THE FORMAT    IPUT1230
C             FOR THE TABULATION IS AS FOLLOWS **                     IPUT1240
C                                                                     IPUT1250
C                                                                     IPUT1260
C     STATN(1 + 4*N ) = LATITUDE RADIANS                              IPUT1270
C          (2 + 4*N ) = LONGITUDE IN RADIANS                          IPUT1280
C          (3 + 4*N ) = ALTITUDE RELATIVE TO REF. ELLIPSOID IN KM     IPUT1290
C          (4 + 4*N ) = STATION NAME (6 CHARACTERS OR LESS )          IPUT1300
C     HORCOR( N )     = HORIZON CORRECTION  (RAD)                     IPUT1310
C     STERR(1 + 3*N ) = VARIANCE IN STATION LATITUDE   RAD**2         IPUT1320
C          (2 + 3*N ) = VARIANCE IN STATION LONGITUDE  RAD**2         IPUT1330
C          (3 + 3*N ) = VARIANCE IN STATION ALTITUDE   KM**2          IPUT1340
C     SNOISE(1+ 4*N ) = VARIANCE IN RANGE        KM**2                IPUT1350
C           (2+ 4*N ) = VARIANCE IN RANGE-RATE (KM/SEC)**2            IPUT1360
C           (3+ 4*N ) = VARIANCE IN AZIMUTH      RAD**2               IPUT1370
C           (4+ 4*N ) = VARIANCE IN ELEVATION    RAD**2               IPUT1380
C                                                                     IPUT1390
C     THESE DATA MUST BE LOADED INTO COMMON STARTING AT LOCATIONS 36, IPUT1400
C             77,87, AND 177 RESPECTIVELY.                            IPUT1410
C                                                                     IPUT1420
C                                                                     IPUT1430
C                                                                     IPUT1440
C     STATION ONE( FLOYD-RADC ) * * * * * * * * * * * * * * * * * * * IPUT1450
C                                                                     IPUT1460
      P(1,1)  = 43.2234  *CONV2                                       IPUT1470
      P(1,2)  = 284.5792 * CONV2                                      IPUT1480
      P(1,3)  =.164                                                   IPUT1490
      P(1,4)  = V1                                                    IPUT1500
      SN(1,1) = 225.E-6                                               IPUT1510
      SN(1,2) = 1.E-8                                                 IPUT1520
```

```
      SN(1,3) = 4.E-6                                                  IPUT1530
      SN(1,4) = 4.E-6                                                  IPUT1540
      SE(1,1) = 1.E-12                                                 IPUT1550
      SE(1,2) = 1.E-12                                                 IPUT1560
      SE(1,3) = 1.E-8                                                  IPUT1570
      HC(1)   = 0.                                                     IPUT1580
C                                                                      IPUT1590
C     STATION TWO( ANTIGUA       * * * * * * * * * * * * * * * * * * * * IPUT1600
C                                                                      IPUT1610
      P(2,1)  = 17.14250  * CONV2                                      IPUT1620
      P(2,2)  = 298.20667 * CONV2                                      IPUT1630
      P(2,3)  = .0457                                                  IPUT1640
      P(2,4)  = V2                                                     IPUT1650
      SN(2,1) = 225.E-6                                                IPUT1660
      SN(2,2) = 1.E-8                                                  IPUT1670
      SN(2,3) = 4.E-6                                                  IPUT1680
      SN(2,4) = 4.E-6                                                  IPUT1690
      SE(2,1) = 1.E-10                                                 IPUT1700
      SE(2,2) = 1.E-10                                                 IPUT1710
      SE(2,3) = 1.E-4                                                  IPUT1720
      HC(2)   = 0.                                                     IPUT1730
C                                                                      IPUT1740
C     STATION THREE( ASCENSION    ) * * * * * * * * * * * * * * * * * * IPUT1750
C                                                                      IPUT1760
      P(3,1)  = - 7.93148 * CONV2                                      IPUT1770
      P(3,2)  =  345.59709* CONV2                                      IPUT1780
      P(3,3)  = 0.2752                                                 IPUT1790
      P(3,4)  = V3                                                     IPUT1800
      SN(3,1) = 225.E-6                                                IPUT1810
      SN(3,2) = 1.E-8                                                  IPUT1820
      SN(3,3) = 4.E-6                                                  IPUT1830
      SN(3,4) = 4.E-6                                                  IPUT1840
      SE(3,1) = 1.E-10                                                 IPUT1850
      SE(3,2) = 1.E-10                                                 IPUT1860
      SE(3,3) = 1.E-4                                                  IPUT1870
      HC(3)   = 0.                                                     IPUT1880
C                                                                      IPUT1890
```

```
C     STATION FOUR ( JOHANESBURG) * * * * * * * * * * * * * * * * * * * IPUT1900
C                                                                       IPUT1910
      P(4,1)  = -25.88765* CCNV2                                        IPUT1920
      P(4,2)  = +27.68478* CONV2                                        IPUT1930
      P(4,3)  = 1.38192                                                 IPUT1940
      P(4,4)  = V4                                                      IPUT1950
      SN(4,1) = 2500.E-6                                                IPUT1960
      SN(4,2) = 4.E-8                                                   IPUT1970
      SN(4,3) = 4.E-8                                                   IPUT1980
      SN(4,4) = 4.E-8                                                   IPUT1990
      SE(4,1) = 1.E-10                                                  IPUT2000
      SE(4,2) = 1.E-10                                                  IPUT2010
      SE(4,3) = 9.E-6                                                   IPUT2020
      HC(4)   = 0.                                                      IPUT2030
C                                                                       IPUT2040
C     STATION FIVE ( JODRELL BANK)* * * * * * * * * * * * * * * * * * * IPUT2050
C                                                                       IPUT2060
      P(5,1)  = 53.045636* CCNV2                                        IPUT2070
      P(5,2)   =+357.69389* CONV2                                       IPUT2080
      P(5,3)  = 0.                                                      IPUT2090
      P(5,4)  = V5                                                      IPUT2100
      SN(5,1) = 225.E-6                                                 IPUT2110
      SN(5,2) = 1.E-8                                                   IPUT2120
      SN(5,3) = 4.E-6                                                   IPUT2130
      SN(5,4) = 4.E-6                                                   IPUT2140
      SE(5,1) = 1.E-10                                                  IPUT2150
      SE(5,2) = 1.E-10                                                  IPUT2160
      SE(5,3) = 1.E-4                                                   IPUT2170
      HC(5)   = 0.                                                      IPUT2180
C                                                                       IPUT2190
C     STATION SIX( WOOMERA AUST) * * * * * * * * * * * * * * * * * *    IPUT2200
C                                                                       IPUT2210
      P(6,1)  = -31.38287 * CCNV2                                       IPUT2220
      P(6,2)  = 136.88507 * CONV2                                       IPUT2230
      P(6,3)  = .1508                                                   IPUT2240
      P(6.4)  = V6                                                      IPUT2250
      SN(6,1) = 2500.E-6                                                IPUT2260
```

```
      SN(6,2) = 4.E-8                                                   IPUT2270
      SN(6,3) = 4.E-8                                                   IPUT2280
      SN(6,4) = 4.E-8                                                   IPUT2290
      SE(6,1) = 1.E-10                                                  IPUT2300
      SE(6,2) = 1.E-10                                                  IPUT2310
      SE(6,3) = 9.E-6                                                   IPUT2320
      HC(6)   = 0.                                                      IPUT2330
C                                                                       IPUT2340
C     STATION SEVEN( AMR - GE    ) * * * * * * * * * * * * * * * * * * * IPUT2350
C                                                                       IPUT2360
      P(7,1)  = 28.278103 *CONV2                                        IPUT2370
      P(7,2)  = 279.418297*CONV2                                        IPUT2380
      P(7,3)  = 0.                                                      IPUT2390
      P(7,4)  = V7                                                      IPUT2400
      SN(7,1) = 225.E-6                                                 IPUT2410
      SN(7,2) = 1.E-8                                                   IPUT2420
      SN(7,3) = 4.E-6                                                   IPUT2430
      SN(7,4) = 4.E-6                                                   IPUT2440
      SE(7,1) = 1.E-10                                                  IPUT2450
      SE(7,2) = 1.E-10                                                  IPUT2460
      SE(7,3) = 1.E-4                                                   IPUT2470
      HC(7)   = 0.                                                      IPUT2480
C                                                                       IPUT2490
C     STATION EIGHT( HAWAII      ) * * * * * * * * * * * * * * * * * * * IPUT2500
C                                                                       IPUT2510
      P(8,1)  = 18.823066 * CONV2                                       IPUT2520
      P(8,2)  = 204.31472 * CONV2                                       IPUT2530
      P(8,3)  = 0.                                                      IPUT2540
      P(8,4)  = V8                                                      IPUT2550
      SN(8,1) = 225.E-6                                                 IPUT2560
      SN(8,2) = 1.E-8                                                   IPUT2570
      SN(8,3) = 4.E-6                                                   IPUT2580
      SN(8,4) = 4.E-6                                                   IPUT2590
      SE(8,1) = 1.E-10                                                  IPUT2600
      SE(8,2) = 1.E-10                                                  IPUT2610
      SE(8,3)= 1.E-4                                                    IPUT2620
      HC(8)   = 0.                                                      IPUT2630
```

```
C                                                                          IPUT2640
C     STATION NINE( BERMUDA     ) * * * * * * * * * * * * * * * * * * * *  IPUT2650
C                                                                          IPUT2660
      P(9,1)  = 32.160973 * CCNV2                                          IPUT2670
      P(9,2)  = 295.29918 * CONV2                                          IPUT2680
      P(9,3)  = 0.                                                         IPUT2690
      P(9,4)  = V9                                                         IPUT2700
      SN(9,1) = 225.E-6                                                    IPUT2710
      SN(9,2) = 1.E-8                                                      IPUT2720
      SN(9,3) = 4.E-6                                                      IPUT2730
      SN(9,4) = 4.F-6                                                      IPUT2740
      SE(9,1) = 1.E-10                                                     IPUT2750
      SE(9,2) = 1.E-10                                                     IPUT2760
      SE(9,3) = 1.E-4                                                      IPUT2770
      HC(9)   = 0.                                                         IPUT2780
C                                                                          IPUT2790
C     STATION TEN( MILLSTCNE HILL)* * * * * * * * * * * * * * * * * * *    IPUT2800
C                                                                          IPUT2810
      P(10,1) = 42.4232  * CCNV2                                           IPUT2820
      P(10,2) = 288.5080 * CGNV2                                           IPUT2830
      P(10,3) = 0.                                                         IPUT2840
      P(10,4) = V10                                                        IPUT2850
      SN(10,1)= 225.E-6                                                    IPUT2860
      SN(10,2)= 1.E-8                                                      IPUT2870
      SN(1C,3)= 4.E-6                                                      IPUT2880
      SN(1C,4)= 4.E-6                                                      IPUT2890
      SE(10,1)= 1.E-10                                                     IPUT2900
      SE(10,2)= 1.E-10                                                     IPUT2910
      SE(10,3)= 1.E-4                                                      IPUT2920
      HC(10)  = 0.                                                         IPUT2930
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    IPUT2940
C                                                                          IPUT2950
C                                                                          IPUT2960
C     AT THIS POINT A DECISION WILL BE MADE AS TO WHICH STATIONS ARE       IPUT2970
C             TO BE CONSIDFRED AND THE APPROPRIATE NUMBERS WILL BF         IPUT2980
C             LOADED INTO COMMCN FOR USE IN THE REST OF THF PROGRAM        IPUT2990
C                                                                          IPUT3000
```

```
C                                                                        IPUT3010
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * IPUT3020
C                                                                        IPUT3030
C                                                                        IPUT3040
C                                                                        IPUT3050
      CALL SICRD(N)                                                      IPUT3055     4
      NUMBER = 0                                                         IPUT3060
      DO 120 I=1,10                                                      IPUT3070
      IF( N(I)-1 ) 120,130,130                                           IPUT3080
  130 K = 4 * NUMBER                                                     IPUT3090
      DO 140 J=1,4                                                       IPUT3100
      L = K + J                                                          IPUT3110
      MM=L-NUMBER                                                        IPUT3120
      STATN(L) = P(I,J)                                                  IPUT3130
      SNOISE(L)= SN(I,J)                                                 IPUT3140
      IF(J-3) 150,150,140                                                IPUT3150
  150 STERR(MM) = SE(I,J)                                                IPUT3160
  140 CONTINUE                                                           IPUT3170
      NUMBER= NUMBER +1                                                  IPUT3180
      HORCOR(NUMBER) = HC(I)                                             IPUT3190
  120 CONTINUE                                                           IPUT3200
C                                                                        IPUT3210
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * IPUT3220
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * IPUT3230
C                                                                        IPUT3240
C     AT THIS POINT THE BALANCE OF THE INPUT DATA WILL BE READ AND       IPUT3250
C              STATION DATA CAN BE MODIFIED.  THE INPUT DATA WILL BE     IPUT3260
C              READ INTO THE FOLLOWING LOCATIONS **                      IPUT3270
C                SAT( 1) = MASS                        (KG)              IPUT3280
C                SAT( 2) = AREA OF CROSS-SECTION    (METERS **2 )        IPUT3290
C                SAT( 3) = DRAG COEFFICIENT                              IPUT3300
C                SAT( 4) = SURFACE REFLECTIVITY                          IPUT3310
C                SAT( 5) = RADIUS VECTOR ( X,Y,Z) IN KM  (FRAME OF DATE)IPUT3320
C                          OR THE ELEMENTS A,E,ORB.INC.  IN KM,--,DEG.   IPUT3330
C                SAT( 8) = VELOCITY VECTOR( X,Y,Z) IN KM/SEC ( FRAME OD)IPUT3340
C                          OR THE ELEMENTS W,OMEGA,THETA IN DEG,DEG,DEG IPUT3350
C                SAT(11) = INTEGRAL NO. OF DAYS SINCE 1950.0             IPUT3360
```

```
C             SAT(12) = FRACTIONAL PART OF A DAY (FIXES TIME OF R,V) IPUT3370
C             SAT(13) = WINDEX**INDEX DEFINING FORMAT OF R AND V     IPUT3380
C                            = 1 FOR CARTESIAN VECTORS               IPUT3390
C                            = 2 OR MORE INPUT ORBITAL ELEMENTS      IPUT3400
C             SAT(14) = CHECK**INDEX USED TO BYPASS CHECKING THE     IPUT3410
C                       TRACKING STATIONS EXCEPT AT TIMES WHEN THERE IPUT3420
C                       IS DATA AVAILABLE                            IPUT3430
C                            = 1. FOR CHECK AT EACH STEP ALONG TTAJ  IPUT3440
C                            = 2. OR MORE TO BYPASS CHECKING         IPUT3450
C             SAT(15) = GONO** INDEX USED TO BYPASS CHECKING THE     IPUT3460
C                       TRACKING STATIONS WHEN DATA IS AVAILABLE     IPUT3470
C                       UNLESS THEY RECORDED THE DATA                IPUT3480
C                            = 1. NO BYPASS                          IPUT3490
C                            = 2. CHECK CNLY RECORDING STATION       IPUT3500
C             STT(70) = COVARIANCE MATRIX FOR R AND V (EPOCH OF T  ) IPUT3510
C                                                                    IPUT3520
C     THIS DONE , THE DATA FOR THE STATICNS CAN BE MCDIFIED AS DESIRED. IPUT3530
C            THIS STEP IS ACCOMPLISHED AS FOLLOWS ** IDENTIFY THE    IPUT3540
C            NUMBER OF THE STATION TO BE ALTERED (N) BY COUNTING     IPUT3550
C            THE NON-ZERO INDICES (N(I)) ON THE FIRST INPUT CARD     IPUT3560
C            BEFORE THE STATICN TO BE CHANGED . WITH                 IPUT3570
C            THIS NUMBER KNOWN , THE LOCATION OF THE DATA FOR THAT   IPUT3580
C            STATION IN THE VARIOUS ARRAYS IS KNOWN AND CAN BE       IPUT3590
C            ALTERED BY REFERRING TO THE DISCUSSION PRECEDING        IPUT3600
C            THE PRESENTATION OF THE STATION DATA .                  IPUT3610
C                                                                    IPUT3620
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * IPUT3630
C                                                                    IPUT3640
  202 CALL REED                                                      IPUT3710    35
      KCHECK = CHECK                                                 IPUT3720
      NOGO = GONO                                                    IPUT3730
      IF(WINDEX-1.) 210,21C,220                                      IPUT3740
  22C A = RVEC(1)                                                    IPUT3750
      E = RVEC(2)                                                    IPUT3760
      CI= RVEC(3)                                                    IPUT3770
      W = VVEC(1)                                                    IPUT3780
      OM= VVEC(2)                                                    IPUT3790
```

```
      TH= VVEC(3)                                                       IPUT3800
      CALL POSVEL( A,E,OI,W,OM,TH )                                     IPUT3810    41
  210 CALL INITAL                                                       IPUT3820    43
      WRITE(    6,300)DJ,DJF,(RVEC(I),I=1,3 ),(VVEC(I),I=1,3),((PP(I,J),JIPUT3830
     1=1,6),I=1,6 ) ,SMASS,AREA,CD,REFLEK                               IPUT3840    44
  300 FORMAT(14H1 INPUT DATA  //15H  DATE(1950)  = 2E17.8/15H  RADIUS VEIPUT3850
     1C. = 3E17.8 / 15H  VELOCITY VEC= 3E17.8 ///27H  INITIAL COVARIANCEIPUT3860
     2 MATRIX // 6E17.8/6E17.8/6E17.8/6E17.8/6E17.8/6E17.8 //           IPUT3870
     316H  SATELLITE DATA /15H  MASS        =E17.8/15H  AREA        =   IPUT3880
     4E17.8 /15H  CD           =E17.8/15H  REFLECTIVITY=E17.8 ////    )  IPUT3890
      MP = 4 * NUMBER                                                   IPUT3900
      WRITE(    6,301) (STATN(K),K=4,MP,4 )                             IPUT3910    67
  301 FORMAT(45H  THE FOLLOWING STATIONS WILL BE CONSIDERED   // 5(10X,AIPUT3920
     16) / 5(10X,A6 ))                                                  IPUT3930
      WRITE(    6,302)                                                  IPUT3931    74
  302 FORMAT(1H1)                                                       IPUT3932
      IF( CODUMP.NE.0. )  CALL DADUMP                                   IPUT3933
C                                                                       IPUT3940
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * IPUT3950
C                                                                       IPUT3960    76
      RETURN                                                            IPUT3970
      END                                                               IPUT3980
```

SUBROUTINE INPUT

COMMON VARIABLES

COMMON BLOCK // ORIGIN 00001 LENGTH 00607

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DATA	00000	R	DATA	00000	R	CON	00000	R
SAT	00017	R	SDA	00043	R	STT	00435	R
WRK	00606	R	RE	00000	R	RPOL	00001	R
COEFJ	00002	R	COFFH	00003	R	COEFD	00004	R
GMERTH	00005	R	OMEGA	00006	R	GMMOON	00007	R
GMSUN	00010	R	AU	00011	R	CONDAY	00012	R
CONV2	00013	R	NIN	00014	I	NOUT	00015	I
SMASS	00017	R	AREA	00020	R	CD	00021	R
REFLEK	00022	R	RVEC	00023	R	VVEC	00026	R
DJ	00031	R	DJF	00032	R	WINDEX	00033	R
CHECK	00034	R	GOND	00035	R	CODUMP	00036	R
KCHECK	00037	I	NOGO	00040	I	STATN	00043	R
HORCOR	00113	R	STERR	00125	R	SNOISE	00257	R
NUMBER	00423	I	PP	00542	R			

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
P	00610	R	SN	00660	R	SE	00730	R
HC	00766	R	N	01000	I			

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
V1	01012	R	V2	01013	R	V3	01014	R
V4	01015	R	V5	01016	R	V6	01017	R
V7	01020	R	V8	01021	R	V9	01022	R

```
      ****                                                                 01/24/86            PAGE  26
            IPUT                                        STORAGE  MAP

V10          01023        R        I           01024       I       K           01025        I
J            01026        I        L           01027       I       MM          01030        I
A            01031        R        E           01032       R       OI          01033        R
W            01034        R        OM          01035       R       TH          01036        R
MP           01037        I

                                                         ENTRY POINTS

     INPUT         SECTION      4

                                                         SUBROUTINES CALLED

     SICRD         SECTICN      5        REED         SECTION     6        POSVEL      SECTION     7
     INITAL        SECTION      8        .FWRD.       SECTION     9        DADUMP      SECTION     10
     .UN06.        SECTICN      11       .FFIL.       SECTION     12       .FCNV.      SECTION     13
     CC.1          SECTION      14       CC.2         SECTION     15       CC.3        SECTION     16
     CC.4          SECTION      17       SYSLOC       SECTION     18

                                                 EFN    IFN    CORRESPONDENCE

EFN          IFN          LOCATION    EFN         IFN          LOCATION    EFN         IFN          LOCATION
120           31A          01772      130          12A          01706      140          26A          01757
150           23A          01752      202          34A          01774      210          42A          02042
220           40A          02015      300         FORMAT        01136      301         FORMAT        01224
302          FORMAT        01241
  DECK LENGTH IN OCTAL IS      01372.
```

2.2.1.1 Subroutine REED

Purpose: REED is designed as a simple reading routine for numeric data and alphabetical characters. This routine is essential to the program if permanent data of the type listed in subroutine INPUT are to be retained or changed as desired

Deck Name: READ

Calling Sequence: CALL REED

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	NIN	-	1	DATA (13)	The input tape number.
I	N	-	1	-	The location in the common array (DATA) at which the floating point number presented in the second field of the card is to be placed. This number is assigned a 12-digit field but should be rightly justified. When N = 999, the routines interprets this as a signal that no more data (numerical) are available and transfers to a second mode. In this mode, the routine reads one 6 character alphabetical name per card into memory until N = 9999. At this time, return is triggered.
I/O	D(K)	DATA	5	DATA (N, N+1, N+2, N+3, N+4)	Up to 5 consecutive pieces of information (floating point numbers) which are to be stored in the common array. If any of the 5 numbers is blank, the balance of the card is not read.

Subroutines Required: None

Functions Required: None

Approximate Deck Length: 130 (decimal)

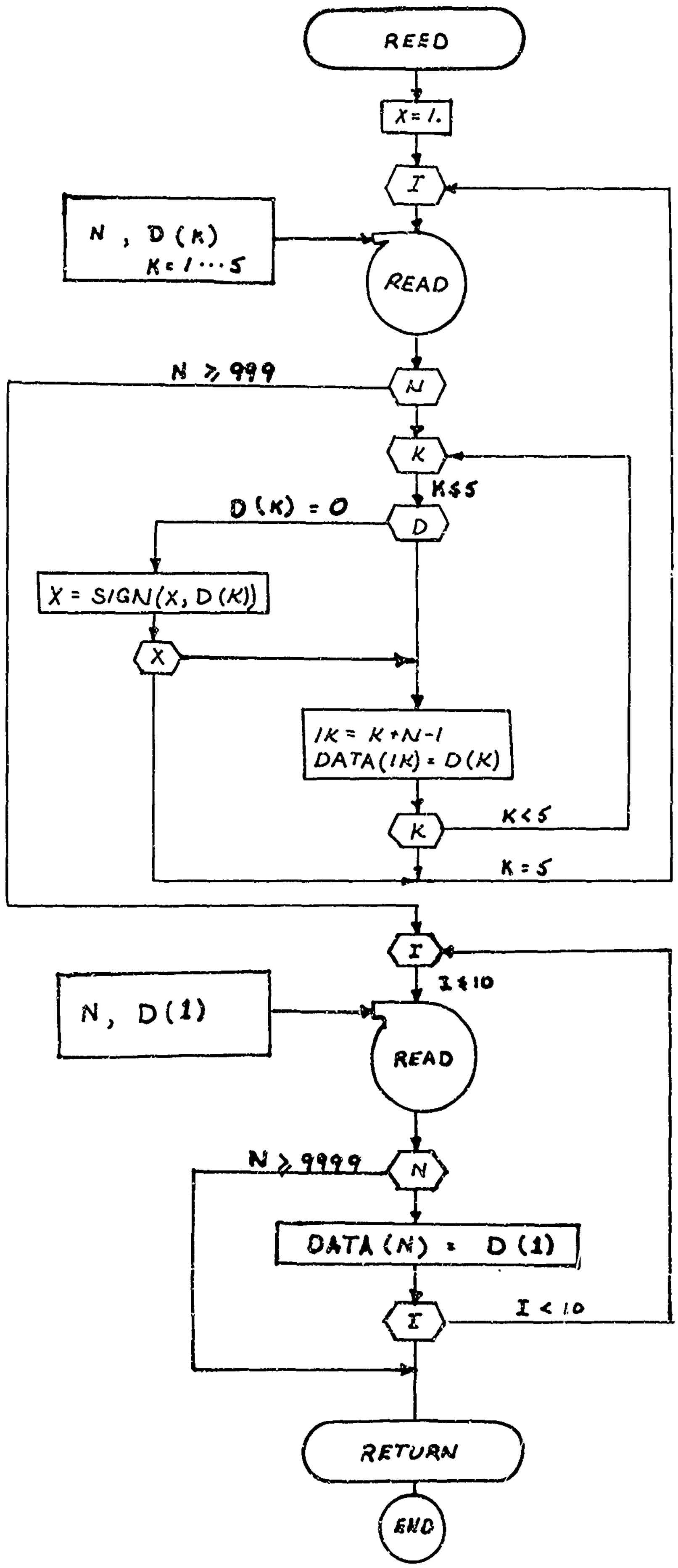
Description:

REED is a semi-general purpose read routine which is designed to accept cards in an I12, 5E 12.8 Format and assign the data recorded in the second through sixth fields of the card to the consecutive locations in common (DATA), beginning with the location read in the first field. Several comments governing the operation of REED are in order, however, since they affect the functioning of REED and thus of the main program. The first is that REED will not search a data card past the first blank field which it recognizes. (Blanks are interpreted as -0.0; thus, zeros must be input using no sign or a plus, e.g., 0.0). If non-consecutive pieces of information are to be placed in the program as data or as new constants, it is necessary to start a new card following the point of discontinuity (or else fill in the data for the intervening location(s)).

The second observation is that all data are assumed to be floating point numbers for the sake of simplicity. Thus, if data are to be utilized as fixed point numbers, a fixing operation must be performed in the calling program after the data have been read. This operation is quite easily accomplished by equating a fixed point name to the floating point variable and equivalenting both variables to locations in the common array (DATA).

The third observation pertains to the method utilized to identify the time at which data are complete. A signal of quite arbitrary nature can be utilized; however, it was felt that the information should be recorded in the first field since this field alone is <u>always</u> read. Further, since the DATA array used for common generally requires fewer than 1000 locations, it was decided that when $N \geq 999$ no more numerical data would be read. <u>A card bearing this number must thus follow all of the data.</u>

When the program identifies a location $N \geq 999$, return to the calling routine is not triggered immediately. Rather, a test is made first to see if any alpha-numeric information is to be read. If $N < 9999$, additional cards are read in a format I12, A6. If $N \geq 9999$, transfer to the calling program is effected. Thus, <u>this card must also be present in the deck whether there are alpha-numeric data or not.</u>


REED
X = 1.
I
N , D (K)
K = 1 ··· 5
READ
N ≥ 999
N
K
K ≤ 5
D (K) = 0
D
X = SIGN (X, D (K))
X
IK = K + N - 1
DATA (IK) = D (K)
K
K < 5
K = 5
I
I ≤ 10
N, D(1)
READ
N ≥ 9999
N
DATA (N) = D (1)
I
I < 10
RETURN
END

```
      SUBROUTINE RFED                                                   READ0020
C                                                                       READ0030
C                                                                       READ0040
C                                                                       READ0050
C     THIS ROUTINE READS A DATA ARRAY AND STORES IT IN COMMON            READ0060
C     THE FORMAT FOR THE DATA REQUIRES THAT THE SUBSCRIPT FOR THE       READ0070
C              FIRST NUMBER BE GIVEN IN THE FIRST 12 -DIGIT FIELD        READ0080
C              5-NUMBERS FOLLOW. IF A BLANK FIELD IS ENCOUNTERED ,      READ0090
C              NO NUMBER AFTER THE BLANK WILL BE RECORDED.              READ0100
C     WHEN THE NUMBER 999 IS ENCOUNTERED IN THE FIRST FIELD OF ANY      READ0110
C              CARD, THE MACHINE WILL INTERPRET IT AS A STOP            READ0120
C              DATA COMMAND AND START READING STATION NAMES COMMAND.    READ0130
C              ALL READING STOPS WHEN 9999 IS READ IN FIELD NO. ONE     READ0140
C                                                                       READ0150
      DIMENSION  CON(1), SAT(1), SDA(1), WRK(1), STT(1)                 READ0160
      COMMON DATA(1)                                                    READ0170
      EQUIVALENCE  (DATA(  1),CON), (DATA( 16),SAT), (DATA( 36),SDA)    READ0180
     2             ,(DATA(286),STT), (DATA(391),WRK)                    READ0190
C                                                                       READ0200
      DIMENSION    D(5)                                                 READ0206
      EQUIVALENCE  (CON(13),NIN)                                        READ0210
C                                                                       READ0220
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   READ0230
C                                                                       READ0240
    1 FORMAT ( I12,5E12.8)                                              READ0250
      X =1.0                                                            READ0260
      DO 100 I=1,1000                                                   READ0270
      READ (  5 ,1) N,(D(K),K=1,5)                                      READ0280    4
      IF( N -999) 200,400,400                                           READ0290
  200 DO 210 K= 1,5                                                     READ0300
      IF(D(K)) 220,230,220                                              READ0310
  230 X = SIGN ( X, D(K) )                                              READ0320
      IF( X) 100,100,220                                                READ0330
  220 IK=K+N-1                                                          READ0340
      DATA(IK) = D(K)                                                   READ0350
  210 CONTINUE                                                          READ0360
```

```
  100 CONTINUE                                                     READ0370
C                                                                  READ0380
C                                                                  READ0390
  400 DO 401 II =1,10                                              READ0400
      READ(  5,402) N,D(1)                                         READ0410     31
      IF( N -9999) 403,500,500                                     READ0420
  403 DATA(N) = D(1)                                               READ0430
  401 CONTINUE                                                     READ0440
  402 FORMAT( I12,A6 )                                             READ0450
  500 CONTINUE                                                     READ0460
      RETURN                                                       READ0650
      END                                                          READ0470
```

```
****
      READ                          STORAGE  MAP          01/14/86              PAGE  34

                              SUBROUTINE  REED

                                 COMMON  VARIABLES

              COMMON BLOCK      //        ORIGIN     00001          LENGTH     00607

SYMBOL    LOCATION    TYPE    SYMBOL    LOCATION    TYPE    SYMBOL    LOCATION    TYPE
DATA       00000       R      DATA       00000       R      CON        00000       R
SAT        00017       R      SDA        00043       R      STT        00435       R
WRK        00606       R      NIN        00014       I

                              DIMENSIONED PROGRAM VARIABLES

SYMBOL    LOCATION    TYPE    SYMBOL    LOCATION    TYPE    SYMBOL    LOCATION    TYPE
D          00610       R

                              UNDIMENSIONED PROGRAM VARIABLES

SYMBOL    LOCATION    TYPE    SYMBOL    LOCATION    TYPE    SYMBOL    LOCATION    TYPE
X          00615       R      I          00616       I      N          00617       I
K          00620       I      IK         00621       I      II         00622       I

                                   ENTRY POINTS

    REED        SECTION     4

                                 SUBROUTINES CALLED

    .FRDD.      SECTION     5        .UNO5.     SECTION     6        .FRTN.     SECTION     7
    .FCNV.      SECTION     8        SYSLOC     SECTION     9
```

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1	FORMAT	00635	100	25A	00721	200	11A	00670
400	28A	00723	210	23A	00717	220	19A	00706
230	16A	00675	401	36A	00751	402	FORMAT	00640
403	34A	00744	500	39A	00753			

DECK LENGTH IN OCTAL IS 00165.

2.2.1.1.1 Subroutine SICRD

Purpose: SICRD is a special purpose subroutine which operates in conjunction with the program logic to assure that the first data card is the station I. D. card. If not, execution is terminated.

Deck Name: SICR

Calling Sequence: CALL SICRD (N)

Input/Output

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
O	N	-	10	Arg	array of numbers indicating which of the ten stations of the program are to be utilized (1 = yes, 0 = no)
I	NIN NOUT	-	1 1	CON (13) CON (14)	input and output tape numbers.

Subroutines Required None
Functions Required None
Approximate Deck Length 100 (Octal)

Discussion:

This routine is designed to read the first data card in an "A-FORMAT" and to test the first word on that card to determine if it is in fact the station I. D. card (SIC). If the card is identified as the SIC, the words in the next 10 fields of the card are tested for zeros. Finally, an array of numbers (N) is constructed of zeros and ones so that the main program will employ the proper network. It is important to note that since the data are read in an A Format <u>any</u> non-zero characters can be employed to identify the users desire to include a station.

If the first card is not the SIC, then execution is terminated with an error message.

```
      SUBROUTINE SICRD(N)                                               SICRD002
C                                                                       SICRD004
C        THIS SUBROUTINE READS THE FIRST DATA CARD AND INSURES THAT     SICRD006
C     IT IS THE STATION IDENTIFICATION CARD (SIC CARD).                 SICRD008
C                                                                       SICRD010
      DIMENSION  N(10),DN(10),IDN(10)                                   SICRD012
C                                                                       SICRD014
      EQUIVALENCE  (SIC,ISIC),(DSIC,IDSIC),(ZERO,IZERO),(DN,IDN)        SICRD016
C                                                                       SICRD018
      DATA  SIC,ZERO /4HSIC ,4H   0/                                    SICRD020
C                                                                       SICRD022
C                                    READ FIRST DATA CARD.              SICRD024
    5 READ(5,100)  DSIC,(DN(I),I=1,10)                                  SICRD026
C                                    TEST FIRST FIELD.  IF CARD WAS     SICRD028
C                                    NOT A SIC CARD, WRITE ERROR        SICRD030
C                                    MESSAGE AND TERMINATE EXECUTION.  SICRD032     1
   10 IF(IDSIC.NE.ISIC) GO TO 40                                        SICRD034
C                                    CARD READ WAS A SIC CARD.  LOAD    SICRD036
C                                    N ARRAY WITH COMPARE AND SET       SICRD038
C                                    TECHNIQUE.                         SICRD040
   15 DO 30  I=1,10                                                     SICRD042
   20 N(I) = 1                                                          SICRD046
   25 IF(IDN(I).EQ.IZERO)  N(I) = 0                                     SICRD048
   30 CONTINUE                                                          SICRD050
      RETURN                                                            SICRD052
C                                    ERROR EXIT                         SICRD056
   40 WRITE(6,110)                                                      SICRD058     25
      CALL DUMP                                                         SICRD060     26
      RETURN                                                            SICRD062
C                                                                       SICRD066
  100 FORMAT(11A4)                                                      SICRD068
  110 FORMAT(1H1,5X,39H***** ERROR PRINT FROM SUBROUTINE SICRD          SICRD070
     2 /1H ,5X,93H      SIC DATA CARD FORMAT NOT CORRECT OR SIC CARD IS SICRD072
     3NOT IN PROPER LOCATION WITHIN DATA DECK   /1H ,5X,39H      EXECUTASICRD076
     4TION TERMINATED BY PROGRAM    )                                   SICRD078
      END                                                               SICRD080
```


****

SUBROUTINE SICRD

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DN	C0004	R	IDN	00004	I			

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
SIC	C0001	R	ISIC	00001	I	DSIC	0C002	R
IDSIC	0C002	I	ZERO	00003	R	IZERO	0C003	I

ENTRY POINTS

SICRD	SECTION	3

SUBROUTINES CALLED

.FRDD.	SECTION	4	.FWRD.	SECTION	5	DUMP	SECTION	6
.UN05.	SECTION	7	.FRTN.	SECTION	8	.FCNV.	SECTION	9
.UN06.	SECTION	10	.FFIL.	SECTION	11	SYSLOC	SECTION	12

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
5	1A	00073	100	FORMAT	00025	10	8A	00110
40	25A	00126	15	11A	00114	30	22A	00123
20	15A	00115	25	17A	00117	110	FORMAT	00026

DECK LENGTH IN OCTAL IS 00146.

2.2.1.2 Subroutine INITAL

Purpose: INITAL is designed to set up the solution to be performed within TRAJ by taking the input data and storing the position and velocity vectors and the associated epoch in cells set aside for the purpose of computing the conic reference trajectory and the state transition matrix.

Deck Name: INIT

Calling Sequence: CALL INITAL

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	R, V	$\vec{r}$, $\vec{v}$	3, 3	SAT (5) SAT (8)	Radius and velocity vectors in the true equator of date frame of reference (Km, Km/sec)
I/O	TW, TF	t_o	1, 1	SAT (11, 12) WRK (50, 51)	Whole and fractional number of mean solar days elapsed since the reference epoch of 1950.0 (JD 2433282.423)
O	ROTATE ROTINV	NP $P^T N^T$	3 X 3 3 X 3	WRK (1) WRK (10)	Transformation matrices relating true equator of date frame of reference to the mean equator of 1950.0 system. ($r_D = NPr_{50}$)
O	RCON VCON	r_c (0) v_c (0)	3 3	WRK (28) WRK (31)	Position and velocity vectors used to describe the conic reference trajectory for the Encke integration (Km, Km/sec)
O	TCON TCONF	t_c	1, 1	WRK (34) WRK (35)	The epoch corresponding to the radius and velocity vectors used to define the conic reference trajectory.

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
0	RTRAN VTRAN	r_{n-1} v_{n-1}	3 3	WRK (36) WRK (39)	Position and velocity vectors utilized to define the state from which errors will propagate to the time of data acquisition. These vectors are in the true equator of date frame. (Km, Km/sec)
0	TTRANW, TTRANF	t_{n-1}	1, 1	WRK (42) WRK (43)	The epoch defining the point from which errors will be propagated
0	R50 V50	$\vec{r}50$ $v50$	3 3	WRK (44) WRK (47)	Position and velocity vectors corresponding to R and V in the frame of 1950.0 (Km, Km/sec)
0	DR DV	$\Delta\vec{r}$ $\Delta\vec{v}$	3 3	WRK (52) WRK (55)	Position and velocity vectors for the satellite relative to the conic reference trajectory (Km, Km/sec)
0	RCONIC VCONIC	$\vec{r}_c$ (t) $\vec{v}_c$ (t)	3 3	WRK (117) WRK (120)	Position and velocity vectors on the conic reference trajectory at the epoch t (Km, Km/sec)
0	STATE	X (t)	6	STT (64)	The initial components of the vector composed of errors in the radius and velocity vectors (Km, Km/sec)

Subroutines Required: EQINOX (relates frame of date to 1950.0)
MATMPY (matrix multiplication)

Functions Required: None

Approximate Deck Length: 134 (octal)

Discussion:

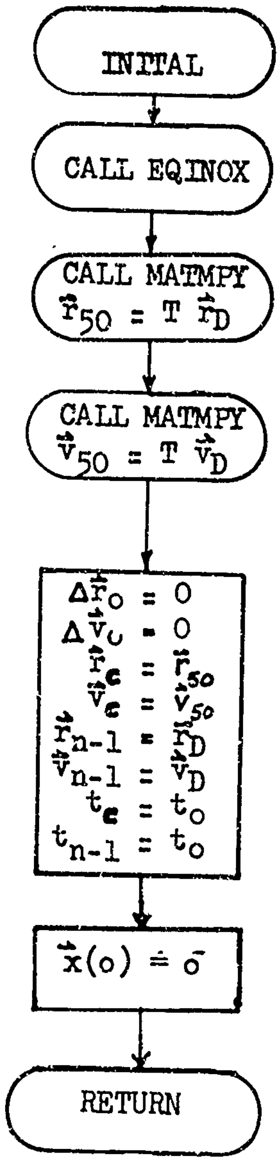
INITAL is intended to take the position and velocity vectors in the true equator of date frame of reference and compute the corresponding position and velocity vectors in the computational frame (the mean equator of 1950.0). This is accomplished by calling EQINOX and performing the following multiplication

$$\vec{r}_{50} = P^T N^T \vec{r}_D$$

$$\vec{v}_{50} = P^T N^T \vec{v}_D$$

Having completed this operation, the vectors required to perform the integration of the trajectory and the computation of the state transition matrix are loaded into common. At this point, the process is completed by establishing the components of the initial state vector (i.e., $\begin{Bmatrix} \delta \vec{r}(0) \\ \delta \vec{v}(0) \end{Bmatrix}$).

Computational Logic:



```
      SUBROUTINE INITAL                                                 INIT0020
C                                                                       INIT0030
C     THIS ROUTINE IS DESIGNED TO OPERATE IN THE INPUT PACKAGE AND IS   INIT0040
C              INTENDED TO PERFORM THE FOLLOWING FUNCTIONS **           INIT0050
C                1) TRANSFORM VECTORS TO THE FRAME OF 1950.0            INIT0060
C                2) ESTABLISH INITIAL CONDITIONS FOR REQUIRED VECTORS   INIT0070
C                                                                       INIT0080
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * INIT0090
C                                                                       INIT0100
      DIMENSION  CON(1), SAT(1), SDA(1), WRK(1), STT(1)                 INIT0110
C                                                                       INIT0120
      COMMON DATA                                                       INIT0130
C                                                                       INIT0140
      EQUIVALENCE  (DATA(  1),CON), (DATA( 16),SAT), (DATA( 36),SDA)    INIT0150
     2             ,(DATA(286),STT), (DATA(391),WRK)                    INIT0160
C                                                                       INIT0170
      DIMENSION  ROTATE(3,3),ROTINV(3,3),RCON(3)    ,VCON(3)            INIT0180
     2,RTRAN(3) ,VTRAN(3)   ,R50(3)      ,V50(3)     ,DR(3)             INIT0190
     3,DV(3)    ,R(3)       ,V(3)        ,STATE(6)   ,RCONIC(3)         INIT0200
     4,VCONIC(3)                                                        INIT0204
C                                                                       INIT0210
      EQUIVALENCE           (SAT(  5),R     ), (SAT(  8),V     )        INIT0220
     2  ,(SAT( 11),TXW   ), (SAT( 12),TXF   )                           INIT0230
C                                                                       INIT0240
      EQUIVALENCE           (WRK(  1),ROTATE), (WRK( 10),ROTINV)        INIT0250
     2  ,(WRK( 28),RCON  ), (WRK( 31),VCON  ), (WRK( 34),TCONW )        INIT0260
     3  ,(WRK( 35),TCONF ), (WRK( 36),RTRAN ), (WRK( 39),VTRAN )        INIT0270
     4  ,(WRK( 42),TTRANW), (WRK( 43),TTRANF), (WRK( 44),R50   )        INIT0280
     5  ,(WRK( 47),V50   ), (WRK( 52),DR    ), (WRK( 55),DV    )        INIT0290
     6  ,(WRK( 50),TW    ), (WRK( 51),TF    ), (WRK(117),RCONIC)        INIT0300
     7  ,(WRK(120),VCONIC)                                              INIT0305
C                                                                       INIT0310
      EQUIVALENCE           (STT( 64),STATE )                           INIT0320
C                                                                       INIT0330
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * INIT0340
C                                                                       INIT0350
```

```
    INIT        -  EFN   SOURCE STATEMENT  -  IFN(S)  -
C                                                                    INIT0360
C     ROTATE INPUT INTO FRAME OF 1950.0 * * * * * * * * * * * * * * * * INIT0370
      TW = TXW                                                       INIT0380
      TF = TXF                                                       INIT0390
      T = TW + TF                                                    INIT0400
      CALL EQIN3X(T)                                                 INIT0410    2
      CALL MATMPY(ROTINV,3,3,R,3,1,R50)                              INIT0420    4
      CALL MATMPY(ROTINV,3,3,V,3,1,V50)                              INIT0430
C                                                                    INIT0440
C                                                                    INIT0450
C     INITIALIZE CONIC QUANTITIES * * * * * * * * * * * * * * * * * * * INIT0460    6
      DO 10 I=1,3                                                    INIT0470
      DR(I) = 0.                                                     INIT0480
      DV(I) = 0.                                                     INIT0490
      RCON(I) = R50(I)                                               INIT0500
      VCON(I) = V50(I)                                               INIT0510
      RCONIC(I) = R50(I)                                             INIT0520
      VCONIC(I) = V50(I)                                             INIT0530
      RTRAN(I) = R(I)                                                INIT0540
   10 VTRAN(I) = V(I)                                                INIT0550
C                                                                    INIT0560
      TTRANW = TW                                                    INIT0570
      TTRANF = TF                                                    INIT0580
      TCONW = TW                                                     INIT0590
      TCONF = TF                                                     INIT0600
C                                                                    INIT0610
      DO  20  I=1,6                                                  INIT0620
   20 STATE(I) = 0.0                                                 INIT0630
C                                                                    INIT0640
      RETURN                                                         INIT0650
      END                                                            INIT0660
```

STORAGE MAP

SUBROUTINE INITAL

COMMON VARIABLES

COMMON BLOCK // ORIGIN OCCO1 LENGTH C1CCO

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DATA	00000	R	DATA	00000	R	CON	00000	R
SAT	00017	R	SDA	00043	R	STT	00435	R
WRK	00606	R	R	00023	R	V	00026	R
TXW	00031	R	TXF	00032	R	ROTATE	00606	R
ROTINV	00617	R	RCON	00641	R	VCON	00644	R
TCONW	00647	R	TCONF	00650	R	RTRAN	00651	R
VTRAN	00654	R	TTRANW	00657	R	TTRANF	00660	R
R50	00661	R	V50	00664	R	DR	00671	R
DV	00674	R	TW	00647	R	TF	00670	R
RCONIC	00772	R	VCONIC	00775	R	STATE	00534	R

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
T	01001	R						

ENTRY POINTS

INITAL SECTION 4

SUBROUTINES CALLED

EQINOX SECTION 5 MATMPY SECTION 6 SYSLOC SECTION 7

EFN IFN CORRESPONDENCE

FS305 INIT

STORAGE MAP

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
10	23A	01063	20	31A	01100			

DECK LENGTH IN OCTAL IS 00122.

2.2.1.3 Subroutine POSVEL

Purpose: POSVEL accepts data in for form of the orbital elements and computes the position and velocity vectors corresponding to the epoch of the elements.

Deck Name: POSVEL

Calling Sequence: CALL POSVEL (GM A, E, OINC, OMEGAW, OMEGA, THETA, R, V)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	GM	μ	1	CON(6)	The product of Newton's gravitational constant and the mass of the Earth. (Km^3/sec^2)
I	A E OINC ØMEGAW OMEGA THETA	a e i ω Ω θ_o	1 1 1 1 1 1	Arg Arg Arg Arg Arg Arg	The semi-major axis, eccentricity, orbital inclination, argument of perigee, longitude of the ascending node and true anomaly of epoch for the elliptical orbit of interest as expressed in the true equator and equinox of date frame of reference (Km, -, deg, deg, deg, deg)
O	R V	$\vec{r}$ $\vec{v}$	3 3	SAT(5) SAT(8)	The position and velocity vectors in the true equator and equinox of date frame of reference (Km, Km/sec) (cartesian coordinates)

Subroutines Required: None

Functions Required:

SIN, SIND	(sine function)
COS, COSD	(cosine function)
SQRT	(square root)
ATAN	(arc tangent)

Approximate Deck Length: 231 (decimal)

Formulation:

The orbital elements define a corresponding set of position and velocity components (in the same coordinate system). POSVEL is designed to compute this equivalent set of numbers by utilizing elliptic formulae. However, since these formulae are reasonably well known, they will not be developed; rather, those employed will be presented.

The elements which are assumed to be available make up a variation of the classic set to be specific.

a - semi-major axis (Km)

e - orbital eccentricity

i - orbital inclination to the true equator of date (deg)

ω - argument of perigee (deg)

Ω - longitude of the ascending node relative to the true vernal equinox of date (deg)

θ_o - true anomaly of epoch (deg). This element is utilized rather than the time of perigee passage due to the fact that it simplifies the resolution process

Thus, the dynamics of the problem is introduced through these elements by employing elliptic equations as follows:

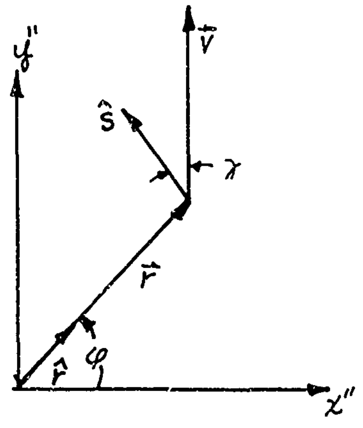
$$r_o = \frac{a(1-e^2)}{1+e\cos\theta_o}$$

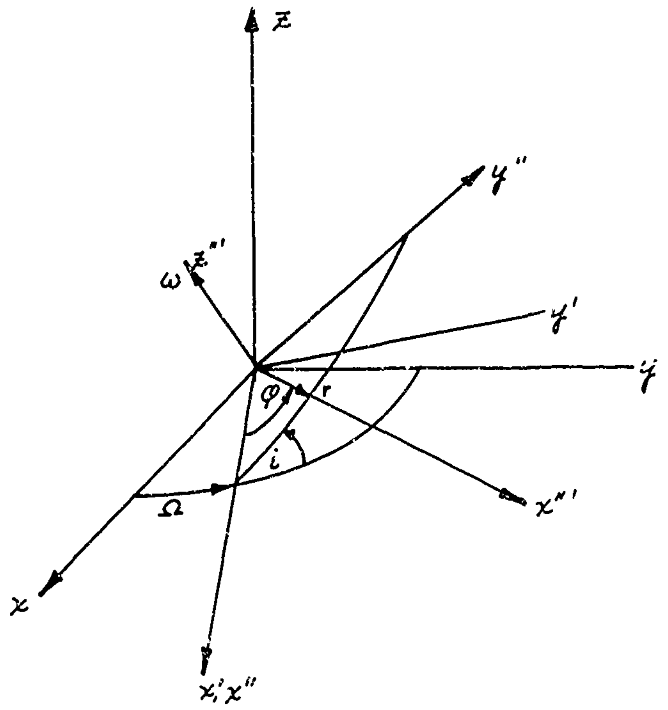
$$v_o = \sqrt{u\left(\frac{2}{r_o} - \frac{1}{a}\right)}$$

$$\gamma_o = \tan^{-1}\left[\frac{e\sin\theta_o}{1+e\cos\theta_o}\right] \qquad -90^\circ \leq \gamma_o \leq +90^\circ$$

where $\gamma_o = \sin^{-1}\left[\frac{\vec{r}_o \cdot \vec{v}_o}{r_o \quad v_o}\right]$

These quantities fail to fully establish the desired information, however, since the orientation of the plane of motion has not been introduced, this step is accomplished by referring to the following sketches.





$\vec{r} = r\,\hat{r}$

$\vec{v} = v \sin\gamma\,\hat{r} + v\cos\gamma\,\hat{s}$

where $\hat{r}$ and $\hat{s}$ are obtained from

$$\begin{Bmatrix} \hat{r} \\ \hat{s} \\ \hat{\omega} \end{Bmatrix} \equiv \begin{Bmatrix} x''' \\ y''' \\ z''' \end{Bmatrix} = T_z(\phi)T_x(i)T_z(\Omega) \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}$$

$$= \begin{bmatrix} \cos\varphi\cos\Omega - \sin\varphi\cos i\sin\Omega & \cos\varphi\sin\Omega + \sin\varphi\cos i\cos\Omega & \sin i\sin\varphi \\ -\sin\varphi\cos\Omega - \cos\varphi\cos i\sin\Omega & -\sin\varphi\sin\Omega + \cos\varphi\cos i\cos\Omega & \cos\varphi\sin i \\ \sin i\sin\Omega & -\sin i\cos\Omega & \cos i \end{bmatrix} \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}$$

and where $T_z(\Omega)$ means the transformation constructed by rotating the coordinates about the Z - axis in a ccw sense through the angle Ω

$T_x(i)$ means the transformation constructed by rotating the coordinates about the x - axis in a ccw sense through the angle i

$\varphi = \theta + \omega$

Computational Logic:

POSVEL

$$\hat{r} = (\cos\varphi\cos\Omega - \sin\varphi\cos i\sin\Omega)\,\hat{x} + (\cos\varphi\sin\Omega + \sin\varphi\cos i\cos\Omega)\,\hat{y} + \sin i\sin\varphi\,\hat{z}$$
$$\hat{s} = (-\sin\varphi\cos\Omega - \cos\varphi\cos i\sin\Omega)\,\hat{x} + (-\sin\varphi\sin\Omega + \cos\varphi\cos i\cos\Omega)\,\hat{y} + \cos\varphi\sin i\,\hat{z}$$

$$r = a(1-e^2)/(1+e\cos\theta)$$
$$v = [\mu(2/r - 1/a)]^{1/2}$$
$$\gamma = \tan^{-1}[e\sin\theta/(1+e\cos\theta)]$$

$$\vec{r} = r\,\hat{r}$$
$$\vec{V} = v\sin\gamma\,\hat{r} + v\cos\gamma\,\hat{s}$$

RETURN

END

```
    PØSV        -  EFN   SØURCE STATEMENT  -  IFN(S)  -

      SUBRØUTINE PØSVEL(A,E,ØINC,ØMEGW,ØMEGA,THETA)                     PØSV0020
C                                                                       PØSV0030
C     THIS RØUTINE COMPUTES THE PØSITIØN AND VELØCITY VECTØRS FRØM      PØSV0040
C          THE CLASSIC ØRBITAL ELEMENT SET.                             PØSV0050
C     GM    = GRAVITATIØN CØNSTANT IN UNITS CØMPATIBLE WITH A,R,ANDV    PØSV0060
C     A     = SEMIMAJØR AXIS                                            PØSV0070
C     E     = ECCENTRICITY                                              PØSV0080
C     ØINC  = ØRBITAL INCLINATIØN    (DEG)                              PØSV0090
C     ØMEGA= RIGHT ASCENSIØN ØF ASCENDING NØDE  (DEG)                   PØSV0100
C     ØMEGW= ARGUMENT ØF PERIAPSE  (DEG)                                PØSV0110
C     THETA= TRUE ANØMALY ØF DATE  (DEG)                                PØSV0120
C     R     = RADIUS VECTØR  ( X,Y,Z )                                  PØSV0130
C     V     = VELØCITY VECTØR ( XD,YD,ZD )                              PØSV0140
C                                                                       PØSV0150
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * PØSV0160
C                                                                       PØSV0170
      DIMENSIØN  CØN(1)     ,SAT(1)                                     PØSV0180
C                                                                       PØSV0190
      CØMMØN DATA                                                       PØSV0200
C                                                                       PØSV0210
      EQUIVALENCE          (DATA(  1),CØN),  (DATA( 16),SAT)            PØSV0220
C                                                                       PØSV0230
      DIMENSIØN            R(3),V(3),S(3)                               PØSV0235
      EQUIVALENCE          (CØN(  6),GM    )                            PØSV0240
      EQUIVALENCE          (SAT(  5),R     ), (SAT(  8),V     )         PØSV0250
C                                                                       PØSV0260
C                                                                       PØSV0270
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * PØSV0280
C                                                                       PØSV0290
C     CØMPUTATIØN ØF UNIT VECTØRS IN LØCAL CØØRDINATES                  PØSV0300
      SI = SIND(ØINC)                                                   PØSV0310    2
      CI = CØSD(ØINC)                                                   PØSV0320    3
      SØ = SIND(ØMEGA)                                                  PØSV0330    4
      CØ = CØSD(ØMEGA)                                                  PØSV0340    5
      PHI= THETA + ØMEGW                                                PØSV0350
      SP = SIND(PHI)                                                    PØSV0360    6
```

```
FS305                                                                11/23/85              PAGE  25
     PØSV         -  EFN   SØURCE STATEMENT  -  IFN(S)  -

      CP = CØSD(PHI)                                                  PØSV0370    7
      R(1) = CØ*CP - SØ*CI*SP                                         PØSV0380
      R(2) = CP*SØ + SP*CI*CØ                                         PØSV0390
      R(3) = SI*SP                                                    PØSV0400
      S(1) =-SP*CØ - CP*CI*SØ                                         PØSV0410
      S(2) =-SP*SØ + CP*CI*CØ                                         PØSV0420
      S(3) = CP*SI                                                    PØSV0430
C                                                                     PØSV0440
C     CØMPUTATIØN ØF RADIUS VELØCITY AND FLIGHT PATH ANGLE            PØSV0450
C                                                                     PØSV0460
      RAD =A*(1.-E*E)/(1.+E*CØSD(THETA))                              PØSV0470    8
      VEL = SQRT( GM *( 2./RAD - 1./A))                               PØSV0480    9
      GAM = ATAN( E*SIND(THETA)/(1.+E*CØSD(THETA)) )                  PØSV0490    10     11     12
      SGV= SIN(GAM) * VEL                                             PØSV0500    13
      CGV= CØS(GAM) * VEL                                             PØSV0510
C                                                                     PØSV0520
C     CØNSTRUCTIØN ØF PØSITIØN AND VELØCITY VECTØRS                   PØSV0530
C                                                                     PØSV0540    14
      DØ 10 I=1,3                                                     PØSV0550
      V(I) = CGV* S(I) + SGV* R(I)                                    PØSV0560
   10 R(I) = RAD* R(I)                                                PØSV0570
      RETURN                                                          PØSV0580
      END                                                             PØSV0590
```

STORAGE MAP

SUBROUTINE POSVEL

COMMON VARIABLES

COMMON BLOCK — ORIGIN 00001 — LENGTH 00031

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DATA	00000	R	DATA	00000	R	CON	00000	R
SAT	00017	R	GM	00005	R	R	00023	R
V	00026	R						

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
S	00032	R						

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
SI	00035	R	CI	00036	R	SO	00037	R
CO	00040	R	PHI	00041	R	SP	00042	R
CP	00043	R	RAD	00044	R	VEL	00045	R
GAM	00046	R	SGV	00047	R	CGV	00050	R

ENTRY POINTS

POSVEL	SECTION	4

SUBROUTINES CALLED

SIND	SECTION	5	COSD	SECTION	6	SQRT	SECTION	7
ATAN	SECTION	8	SIN	SECTION	9	COS	SECTION	10

SYSLOC SECTION 11

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
10	21A	00312						

DECK LENGTH IN OCTAL IS 00333.

2.2.1.4 Block DATA

Purpose: BLOCK stores atmospheric density and lapse rate data and solar-lunar ephemeris data into cells utilized by subroutines ATMS and EPHEM, respectively. BLOCK is executed at the time the program is loaded; thus, the routine is never called in the program.

Deck Name: BLOCK

Calling Sequence: None

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
0	ALT 1 ALT 2 ALT 3	h_1 h_2 h_3	1 1 1	ATCON (1) ATCON (2) ATCON (3)	the altitudes in Km corresponding to the ranges over which lapse rate and density data for the 1962 U.S. standard atmosphere will be quoted.
0	REQT RPOL	R_e R_p	1 1	ATCON (4) ATCON (5)	the equatorial and polar radii of the earth in Km
0	STEP 1 STEP 2	Δh_1 Δh_2	1 1	ATCON (6) ATCON (7)	the intervals at which density data are quoted (between h_1 and h_2, and h_2 and h_3, respectively)
0	RHOF RATE	ρ K	21 21	TABLE (1) TABLE (22)	the density and lapse rate for the 1962 U.S. standard atmosphere at the tabulated intervals
0	EPHAM	t_o Δt_s Δt_m $\vec{r}_m, \vec{v}_m$ $\vec{r}_s, \vec{v}_s$	1 1 1 270 66	EPHOM (1) EPHOM (2) EPHOM (3) EPHOM (4) EPHOM (274)	the date relative to 1950.0 (J.D. 2433282.423) at which the tabulated data begin (days), the time interval for solar ephemeris entries (days), the time interval for lunar ephemeris entries(days), position and velocity vectors for the moon (Km, Km/sec), and position and velocity vectors for the sun (km, Km/sec)

Subroutines Required:	None
Functions Required:	None
Approximate Deck Length:	1 (octal)

```
      BLOCK DATA                                                        BLOK0020
C                                                                       BLOK0030
C     THESE CARDS LOAD ATMOSPHERE ( U.S STANDARD - 1962 )               BLOK0040
C             AND PROVIDE POSITION AND VELOCITY DATA FOR THE SUN        BLOK0050
C             AND MOON IN THE FRAME OF 1950.0                           BLOK0060
C                                                                       BLOK0070
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   BLOK0080
C                                                                       BLOK0090
      COMMON  /TABLE/ RHOF(21), RATE(21)                                BLOK0100
      COMMON /ATCON/ ALT1, ALT2, ALT3, REQT, RPOL, STEP1, STEP2          BLOK0110
      COMMON / EPHOM / EPHAM(339)                                       BLOK0120
C                                                                       BLOK0130
      DATA ALT1, ALT2, ALT3 /100., 200., 700./, REQT /6378.166/,        BLOK0140
     1 RPOL /6356.784/, STEP1, STEP2  /10., 50./                        BLOK0150
C                                                                       BLOK0160
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   BLOK0170
C                                                                       BLOK0180
C     * * * * * * * * * * * * ATMOSPHERE * * * * * * * * * * * * * * * *BLOK0190
C                                                                       BLOK0200
      DATA(RHOF(I),I=1,21) /4.974E-7  ,9.829E-8  ,2.436E-8  ,7.589E-9  ,BLOK0210
     1           3.394E-9  ,1.836E-9  ,1.159E-9  ,8.036E-10 ,5.858E-10 ,BLOK0220
     2           4.347E-10 ,3.318E-10 ,9.975E-11 ,3.585E-11 ,1.465E-11 ,BLOK0230
     3           6.498E-12 ,3.122E-12 ,1.577E-12 ,8.430E-13 ,4.640E-13 ,BLOK0240
     4           2.645E-13 ,1.537E-13 /                                 BLOK0250
C                                                                       BLOK0260
      DATA(RATE(I),I=1,21) /.1725     ,.1541     ,.1305     ,.09448    ,BLOK0270
     1           .06924    ,.05256    ,.04098    ,.03385    ,.03055    ,BLOK0280
     2           .02829    ,.02592    ,.02205    ,.01925    ,.0170     ,BLOK0290
     3           .0154     ,.0141     ,.0130     ,.0122     ,.0115     ,BLOK0300
     1           .0110     ,.0105     /                                 BLOK0310
C                                                                       BLOK0320
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   BLOK0330
C                                                                       BLOK0340
C     * * * * * * * * * * * * EPHEMERIS  * * * * * * * * * * * * * * * *BLOK0350
C                                                                       BLOK0360
      DATA ( EPHAM(I),I=1,3)/                                           BLOK0370
```

```
1 5479.077      , 0.36525000E 03, 0.84000000E 02  /                BLOK0380
 DATA ( EPHAM(I),I=4,63)/                                          BLOK0390
1-0.70247782E 05,-0.36277776E 06,-0.15524963E 06, 0.93949377E 00,  BLOK0400
1-0.12989666E-00,-0.15241475E-00, 0.10333098E 06,-0.34694722E 06,  BLOK0410
1-0.16881503E 06, 0.91933889E 00, 0.25536405E-00, 0.29838797E-01,  BLOK0420
1 0.25263692E 06,-0.27182027E 06,-0.15102310E 06, 0.73395690E 00,  BLOK0430
1 0.58106141E 00, 0.19864140E-00, 0.36125240E 06,-0.13481987E 06,  BLOK0440
1-0.96168648E 05, 0.37440655E-00, 0.82113539E 00, 0.35452610E-00,  BLOK0450
1 0.39081258E 06, 0.13581862E 05,-0.29626390E 05,-0.46969695E-01,  BLOK0460
1 0.87481923E 00, 0.42590836E-00, 0.35369990E 06, 0.16392259E 06,  BLOK0470
1 0.50614960E 05,-0.45504498E-00, 0.75901563E 00, 0.40907497E-00,  BLOK0480
1 0.22723512E 06, 0.28096502E 06, 0.12067740E 06,-0.83539481E 00,  BLOK0490
1 0.47992978E-00, 0.30956405E-00, 0.77854235E 05, 0.33005320E 06,  BLOK0500
1 0.15949935E 06,-0.10095771E 01, 0.96623331E-01, 0.12544969E-00,  BLOK0510
1-0.10628679E 06, 0.31802250E 06, 0.17084964E 06,-0.96010856E 00,  BLOK0520
1-0.32192053E-00,-0.93288352E-01,-0.25836665E 06, 0.21720380E 06,  BLOK0530
1 0.12842447E 06,-0.72258838E 00,-0.71471968E 00,-0.31954692E-00 / BLOK0540
 DATA ( EPHAM(I),I=64,123)  /                                      BLOK0550
1-0.33911225E 06, 0.97408102E 05, 0.72246015E 05,-0.30538394E-00,  BLOK0560
1-0.90996324E 00,-0.45848799E-00,-0.35736063E 06,-0.73888467E 05,  BLOK0570
1-0.19200481E 05, 0.21737552E-00,-0.90127873E 00,-0.48440940E-00,  BLOK0580
1-0.26354197E 06,-0.21550027E 06,-0.10237455E 06, 0.70125832E 00,  BLOK0590
1-0.70896224E 00,-0.41290416E-00,-0.14020613E 06,-0.29462018E 06,  BLOK0600
1-0.15083277E 06, 0.98178807E 00,-0.35353766E-00,-0.22947904E-00,  BLOK0610
1 0.51916093E 05,-0.32225630E 06,-0.17622698E 06, 0.10248832E 01,  BLOK0620
1 0.10337698E-00, 0.21878581E-01, 0.21670982E 06,-0.25743604E 06,  BLOK0630
1-0.14462350E 06, 0.88147689E 00, 0.50785363E 00, 0.24928647E-00,  BLOK0640
1 0.32994590E 06,-0.15441491E 06,-0.89796256E 05, 0.52357153E 00,  BLOK0650
1 0.77944385E 00, 0.41587184E-00, 0.38724329E 06, 0.51500655E 04,  BLOK0660
1-0.24645324E 04, 0.43023048E-01, 0.86729404E 00, 0.47029623E-00,  BLOK0670
1 0.34606686E 06, 0.14889444E 06, 0.81272077E 05,-0.38263620E-00,  BLOK0680
1 0.80509757E 00, 0.44038863E-00, 0.24899276E 06, 0.26817318E 06,  BLOK0690
1 0.14632411E 06,-0.73778787E 00, 0.56510618E 00, 0.30523336E-00/  BLOK0700
 DATA ( EPHAM(I),I=124,183)/                                       BLOK0710
1 0.83658529E 05, 0.34222988E 06, 0.18756323E 06,-0.93178824E 00,  BLOK0720
1 0.20236531E-00, 0.96013178E-01,-0.83417735E 05, 0.34318624E 06,  BLOK0730
1 0.18475436E 06,-0.94047285E 00,-0.15410256E-00,-0.10090744E-00,  BLOK0740
```

```
      BLØCK       -  EFN   SØURCE STATEMENT  -  IFN(S)  -

1-0.24280813E 06, 0.28451353E 06, 0.14587846E 06,-0.76346969E 00,  BLØK0750
1-0.50196432E 00,-0.29503027E-00,-0.35322856E 06, 0.16574296E 06,  BLØK0760
1 0.77413162E 05,-0.42146812E-00,-0.75553752E 00,-0.42141203E-00,  BLØK0770
1-0.39651489E 06, 0.26637565E 05,-0.39249516E 04,-0.21289010E-01,  BLØK0780
1-0.84730782E 00,-0.45281094E-00,-0.35937005E 06,-0.13385931E 06,  BLØK0790
1-0.88453533E 05, 0.42582919E-00,-0.78650998E 00,-0.39822569E-00,  BLØK0800
1-0.25006150E 06,-0.25083090E 06,-0.14577611E 06, 0.79534364E 00,  BLØK0810
1-0.55826461E 00,-0.24595345E-00,-0.10432597E 06,-0.32568986E 06,  BLØK0820
1-0.17664155E 06, 0.96806454E 00,-0.21390165E-00,-0.51697154E-01,  BLØK0830
1 0.89339991E 05,-0.32690211E 06,-0.16093186E 06, 0.9850241  00,  BLØK0840
1 0.22131915E-00, 0.18203875E-00, 0.23193926E 06,-0.25528318E 06,  BLØK0850
1-0.11349309E 06, 0.77241178E 00, 0.61895438E 00, 0.37360413E-00/  BLØK0860
 DATA ( EPHAM(I),I=184,243)/                                       BLØK0870
1 0.33846556E 06,-0.14022343E 06,-0.43232025E 05, 0.36139473E-00,  BLØK0880
1 0.85684197E 00, 0.45607702E-00, 0.35577565E 06, 0.35941610E 05,  BLØK0890
1 0.47709423E 05,-0.15414876E-00, 0.94720755E 00, 0.45793529E-00,  BLØK0900
1 0.29113484E 06, 0.16961060E 06, 0.10721744E 06,-0.62300035E 00,  BLØK0910
1 0.82044052E 00, 0.34327192E-00, 0.16241336E 06, 0.28829765E 06,  BLØK0920
1 0.15387631E 06,-0.93357997E 00, 0.45748128E-00, 0.13442028E-00,  BLØK0930
1-0.32975623E 05, 0.32888559E 06, 0.15138192E 06,-0.10481597E 01,  BLØK0940
1 0.45799801E-03,-0.95959724E-01,-0.17999332E 06, 0.29064213E 06,  BLØK0950
1 0.11759706E 06,-0.93631189E 00,-0.41817886E-00,-0.28599827E-00,  BLØK0960
1-0.31979252E 06, 0.19211543E 06, 0.56813307E 05,-0.56530631E 00,  BLØK0970
1-0.74703696E 00,-0.39262024E-00,-0.37682313E 06, 0.30296557E 05,  BLØK0980
1-0.24374177E 05,-0.11570158E-00,-0.91091619E 00,-0.41945396E-00,  BLØK0990
1-0.35316122E 06,-0.11782046E 06,-0.85392280E 05, 0.32722481E-00,  BLØK1000
1-0.88502721E 00,-0.35651777E-00,-0.25886545E 06,-0.26273017E 06,  BLØK1010
1-0.13918605E 06, 0.70575023E 00,-0.63947488E 00,-0.20390762E-00/  BLØK1020
 DATA ( EPHAM(I),I=244,303)/                                       BLØK1030
1-0.10309367E 06,-0.34681572E 06,-0.15615643E 06, 0.91856205E 00,  BLØK1040
1-0.30382188E-00,-0.40665378E-01, 0.60709404E 05,-0.36537232E 06,  BLØK1050
1-0.14488880E 06, 0.95543024E 00, 0.82755691E-01, 0.12808283E-00,  BLØK1060
1 0.23024838E 06,-0.31086902E 06,-0.10572580E 06, 0.78148480E 00,  BLØK1070
1 0.48444480E-00, 0.26952671E-00, 0.34273709E 06,-0.19922920E 06,  BLØK1080
1-0.47010973E 05, 0.47370192E-00, 0.75438099E 00, 0.34337316E-00,  BLØK1090
1 0.39732704E 06,-0.48334468E 05, 0.15205119E 05, 0.79005685E-01,  BLØK1100
1 0.88805860E 00, 0.35509171E-00, 0.25969815E 08,-0.13283711E 09,  BLØK1110
```

```
    BLØCK      -  EFN   SØURCE STATEMENT  -  IFN(S)  -

1-0.57604471E 08, 0.29808135E 02, 0.49201572E 01, 0.21335439E 01,  BLØK1120
1 0.25947898E 08,-0.13284121E 09,-0.57606132E 08, 0.29809032E 02,  BLØK1130
1 0.49163177E 01, 0.21318875E 01, 0.25926496E 08,-0.13284551E 09,  BLØK1140
1-0.57607681E 08, 0.29809216E 02, 0.49136171E 01, 0.21306460E 01,  BLØK1150
1 0.25917875E 08,-0.13284449E 09,-0.57606748E 08, 0.29810207E 02,  BLØK1160
1 0.49118402E 01, 0.21297985E 01, 0.25901888E 08,-0.13284796E 09,  BLØK1170
1-0.57608023E 08, 0.29810709E 02, 0.49080602E 01, 0.21281333E 01/  BLØK1180
 DATA ( EPHAM(I),I=304,339)/                                       BLØK1190
1 0.25891495E 08,-0.13285235E 09,-0.57609232E 08, 0.29810368E 02,  BLØK1200
1 0.49056256E 01, 0.21270791E 01, 0.25874168E 08--0.13285080E 09,  BLØK1210
1-0.57608055E 08, 0.29812188E 02, 0.49034609E 01, 0.21261606E 01,  BLØK1220
1 0.25848795E 08,-0.13285182E 09,-0.57608315E 08, 0.29813723E 02,  BLØK1230
1 0.48991868E 01, 0.21243198E 01, 0.25836263E 08,-0.13285442E 09,  BLØK1240
1-0.57609329E 08, 0.29814070E 02, 0.48967008E 01, 0.21232317E 01,  BLØK1250
1 0.25809715E 08,-0.13286313E 09,-0.57612987E 08, 0.29813968E 02,  BLØK1260
1 0.48921731E 01, 0.21212412E 01, 0.25787921E 08,-0.13287047E 09,  BLØK1270
1-0.57615668E 08, 0.29813546E 02, 0.48887521E 01, 0.21196672E 01/  BLØK1280
 END                                                               BLØK1290
```

```
    BLØCK                                           STØRAGE  MAP

                                                     BLØCK  DATA

                                               CØMMØN   VARIABLES

                 CØMMØN BLØCK     TABLE            ØRIGIN       0CC00            LENGTH       00052

SYMBØL       LØCATIØN      TYPE   SYMBØL      LØCATIØN      TYPE      SYMBØL      LØCATIØN      TYPE
RHØF          00000          R    RATE         00025          R

                 CØMMØN BLØCK     ATCØN            ØRIGIN       00052            LENGTH       00007
ALT1          00000          R    ALT2         00001          R       ALT3         00002          R
REQT          00003          R    RPØL         00004          R       STEP1        00005          R
STEP2         00006          R

                 CØMMØN BLØCK     EPHØM            ØRIGIN       00061            LENGTH       00523
EPHAM         00000          R
  DECK LENGTH IN ØCTAL IS      00001.
```

2.2.1.5 SUBROUTINE DADUMP

Purpose: DADUMP prints the entire contents of the unlabeled COMMON region DATA. A complete description of the output data may be found in Appendix (COMMON map).

Deck Name: DADMP

Calling Sequence: CALL DADUMP

Input/Output

I/O	FORTRAN Name	Dimension	COMMON/ Argument	Definition
I/O	DATA	525	COMMON	Unlabeled COMMON region composed of the subarrays CON, SAT, SDA, STT, WRK.
I	NOUT	1	CON(17)	Output tape number

Subroutines Required: None

Functions Required: None

Approximate Deck Length: 533 (octal)

Discussion:

The principal purpose of this routine is to provide a print-out of the pertinent conditions at program initiation. After reading the program input data and performing any necessary initialization, subroutine DADUMP is mechanized from subroutine INPUT by setting the flag CODUMP to a non-zero value (input data location 31).

Utilization of this option provides a permanent record of all of the numbers associated with a particular run (gravitational constants, etc.), and a convenient means of checking for system incompatibilities during program checkout.

```
      SUBROUTINE DADUMP                                                        DDMP0020
C                                                                              DDMP0030
C         THIS SUBROUTINE PRINTS THE ENTIRE CONTENTS OF THE COMMON            DDMP0040
C         ARRAY DATA.                                                          DDMP0050
C                                                                              DDMP0060
      DIMENSION  CON(15), SAT(20), SDA(250), STT(105), WRK(135)               DDMP0070
C                                                                              DDMP0080
      COMMON DATA                                                              DDMP0090
C                                                                              DDMP0100
      EQUIVALENCE      (DATA(  1),CON),   (DATA( 16),SAT)                      DDMP0110
     2                ,(DATA( 36),SDA),   (DATA(286),STT)                      DDMP0120
     3                ,(DATA(391),WRK)                                         DDMP0130
C                                                                              DDMP0140
C  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *        DDMP0150
C                                                                              DDMP0160
C                                      WRITE CON SUBARRAY.                     DDMP0170
    5 ND = 1                                                                   DDMP0190
   10 NCON = 1                                                                 DDMP0200
   15 WRITE(6,1005)                                                            DDMP0210     3
   20 DO 50  I=1,3                                                             DDMP0220
   25 NCONE = NCON + 5                                                         DDMP0230
   30 IF( I.EQ.3 )  NCONE = 15                                                 DDMP0240
   35 WRITE(6,1000)  ND, NCON, (CON(J),J=NCON,NCONE )                          DDMP0250    12
   40 NCON = NCONE + 1                                                         DDMP0260
   45 ND = ND + 6                                                              DDMP0270
   50 CONTINUE                                                                 DDMP0280
C                                                                              DDMP0290
C                                      WRITE SAT SUBARRAY.                     DDMP0300
   55 ND = 16                                                                  DDMP0310
   60 NSAT = 1                                                                 DDMP0320
   65 WRITE(6,1010)                                                            DDMP0330    24
   70 DO 100  I=1,4                                                            DDMP0340
   75 NSATE = NSAT + 5                                                         DDMP0350
   80 IF( I.EQ.4 )  NSATE = 20                                                 DDMP0360
   85 WRITE(6,1000)  ND, NSAT, (SAT(J),J=NSAT,NSATE )                          DDMP0370    33
   90 NSAT = NSATE + 1                                                         DDMP0380
```

```
  95 ND = ND + 6                                                      DDMP0390
 100 CONTINUE                                                         DDMP0400
C                                                                     DDMP0410
C                              WRITE SDA SUBARRAY.                    DDMP0420
 105 ND = 36                                                          DDMP0430
 110 NSDA = 1                                                         DDMP0440
 115 WRITE(6,1013)                                                    DDMP0450    45
 120 DO 150  I=1,42                                                   DDMP0460
 125 NSDAE = NSDA + 5                                                 DDMP0470
 130 IF( I.EQ.42 )  NSDAE = 250                                       DDMP0480
 132 IF( I.EQ.30)  WRITE(6,1015)                                      DDMP0490    55
 135 WRITE(6,1000)  ND, NSDA, (SDA(J),J=NSDA,NSDAE )                  DDMP0500    57
 140 NSDA = NSDAE + 1                                                 DDMP0510
 145 ND = ND + 6                                                      DDMP0520
 150 CONTINUE                                                         DDMP0530
C                                                                     DDMP0540
C                              WRITE STT ARRAY.                       DDMP0550
 155 ND = 286                                                         DDMP0560
 160 NSTT = 1                                                         DDMP0570
 165 WRITE(6,1020)                                                    DDMP0580    69
 170 DO 200  I=1,18                                                   DDMP0590
 175 NSTTE = NSTT + 5                                                 DDMP0600
 180 IF( I.EQ.18 )  NSTTE = 105                                       DDMP0610
 185 WRITE(6,1000)  ND, NSTT, (STT(J),J=NSTT,NSTTE )                  DDMP0620    78
 190 NSTT = NSTTE + 1                                                 DDMP0630
 195 ND = ND + 6                                                      DDMP0640
 200 CONTINUE                                                         DDMP0650
C                                                                     DDMP0660
C                              WRITE WRK SUBARRAY.                    DDMP0670
 205 ND = 391                                                         DDMP0680
 210 NWRK = 1                                                         DDMP0690
 215 WRITE(6,1025)                                                    DDMP0700    90
 220 DO 250  I=1,23                                                   DDMP0710
 225 NWRKE = NWRK + 5                                                 DDMP0720
 230 IF( I.EQ.23 )  NWRKE = 135                                       DDMP0730
 235 WRITE(6,1000)  ND, NWRK, (WRK(J),J=NWRK,NWRKE )                  DDMP0740    99
 240 NWRK = NWRKE + 1                                                 DDMP0750
```

```
 245 ND = ND + 6                                                      DDMP0760
 250 CONTINUE                                                         DDMP0770
C                                                                     DDMP0780
 255 CONTINUE                                                         DDMP0790
     WRITE(6,1002)                                                    DDMP0795   109
     RETURN                                                           DDMP0800
C                                                                     DDMP0810
1000 FORMAT(1H ,I4,I5,6E16.8)                                         DDMP0820
1002 FORMAT(1H1)                                                      DDMP0830
1005 FORMAT(1H ,41X,18H*** DATA ARRAY *** /1H , 9HDATA  CON  )        DDMP0840
1010 FORMAT(1H , 9H       SAT )                                       DDMP0850
1013 FORMAT(1H , 9H       SDA  )                                      DDMP0860
1015 FORMAT(1H1,41X,18H*** DATA ARRAY *** /1H , 9HDATA  SDA )         DDMP0870
1020 FORMAT(1H , 9H       STT )                                       DDMP0880
1025 FORMAT(1H1,41X,18H*** DATA ARRAY *** /1H , 9HDATA  WRK )         DDMP0890
     END                                                              DDMP0900
```

SUBROUTINE DADUMP

COMMON VARIABLES

COMMON BLOCK // ORIGIN 00001 LENGTH 01015

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DATA	00000	R	DATA	00000	R	CON	00000	R
SAT	00017	R	SDA	00043	R	STT	00435	R
WRK	00606	R						

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
ND	01016	I	NCON	01017	I	I	01020	I
NCONF	01021	I	NSAT	01022	I	NSATE	01023	I
NSDA	01024	I	NSDAE	01025	I	NSTT	01026	I
NSTTF	01027	I	NWRK	01030	I	NWRKE	01031	I

ENTRY POINTS

DADUMP SECTION 4

SUBROUTINES CALLED

.FWRD.	SECTION	5	.UN06.	SECTION	6	.FFIL.	SECTION	7
.FCNV.	SECTION	8	SYSLOC	SECTION	9			

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
5	1A	01135	10	2A	01137	15	3A	01141

****	DADMP			STØRAGE	MAP	01/22/86		PAGE 39
1005	FØRMAT	01066	20	4A	01147	50	19A	01212
25	8A	01152	30	9A	01155	35	12A	01162
1000	FØRMAT	01062	40	17A	01204	45	18A	01207
55	22A	01214	60	23A	01216	65	24A	01220
1010	FØRMAT	01077	70	25A	01226	100	40A	01271
75	29A	01231	80	30A	01234	85	33A	01241
90	38A	01263	95	39A	01266	105	43A	01273
110	44A	01275	115	45A	01277	1013	FØRMAT	01103
120	46A	01305	150	64A	01361	125	50A	01310
130	51A	01313	132	54A	01320	1015	FØRMAT	01107
135	57A	01331	140	62A	01353	145	63A	01356
155	67A	01363	160	68A	01365	165	69A	01367
1020	FØRMAT	01120	170	70A	01375	200	85A	01440
175	74A	01400	180	75A	01403	185	78A	01410
190	83A	01432	195	84A	01435	205	88A	01442
210	89A	01444	215	90A	01446	1025	FØRMAT	01124
220	91A	01454	250	106A	01517	225	95A	01457
230	96A	01462	235	99A	01467	240	104A	01511
245	105A	01514	255	109A	01521	1002	FØRMAT	01065

DECK LENGTH IN ØCTAL IS 00533.

2.3 THE TRAJECTORY GROUP

The second of the major blocks included within the differential corrections program is the trajectory program. This group of routines evaluates the acceleration vector, integrates the acceleration to define the position and velocity and computes the data relative to the selected tracking stations. These functions are discharged by four separate driver routines.

1)	CONIC	(Generates conic reference trajectory for an Encke solution to the equations of motion - called from INGRAT)
2)	MOTION	(Generates the total acceleration vector for motion relative to the reference ellipse)
3)	INGRAT	(Integrates the acceleration vector obtained from MOTION and defines the position and velocity vectors)
4)	TRAK	(Defines the position and velocity vectors relative to the tracking stations and computes the predicted values of the observed data)

which are mechanized to function in conjunction with a master routine TRAJ. This master is discussed on the following pages and the manner in which these routines are incorporated into the total computational problem is demonstrated.

2.3.1 Subroutine TRAJ

Purpose: TRAJ mechanizes the working portion of the program and serves as the means whereby the reference trajectory is computed and the tracking stations of the problem checked. Once entered, control is not returned to MAIN until the final data point is processed.

Deck Name: TRAJEC

Calling Sequence: CALL TRAJ

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	TCONW TCONF	t_{conic}	1 1	WRK(34) WRK(35)	TW and TF corresponding to the epoch at which the reference ellipse was defined (days).
I	RVEC VVEC	$\vec{r}$ $\vec{v}$	3 3	WRK(44) WRK(47)	The position and velocity vectors in the frame of 1950.0 (Km, Km/sec.)
I	TW TF	t	1 1	WRK(50) WRK(51)	Whole & fractional part of the number of days since 1950.0 for present epoch (Days from J.D. 2433283.423)
I	R RD	$\Delta \vec{r}$ $\Delta \vec{v}$	3 3	WRK(52) WRK(55)	Position and velocity vectors defining the motion relative to the Encke ellipse (Km, Km/sec.)
O	TIME	$t-t_o$	1	WRK(104)	time in seconds since last reference trajectory rectification
O	RCONIC VCONIC	$\vec{r}_c$ $\vec{v}_c$	3 3	WRK(117) WRK(120	Position and velocity vectors at TW plus TF for the conic reference trajectory (Km, Km/sec.)

Subroutines Required:	EPHEM (ephemeris for Sun and Moon position vectors)
	MOTION (driver for evaluating the total acceleration experienced by vehicle)
	INGRAT (numerical integration driver)
	EQINOX (routine for relating the true equator of date to the mean equator of 1950.0)
	TRAK (routine designed to compute topocentric data for tracking stations)
Functions Required:	None
Approximate Deck Length:	115 octal

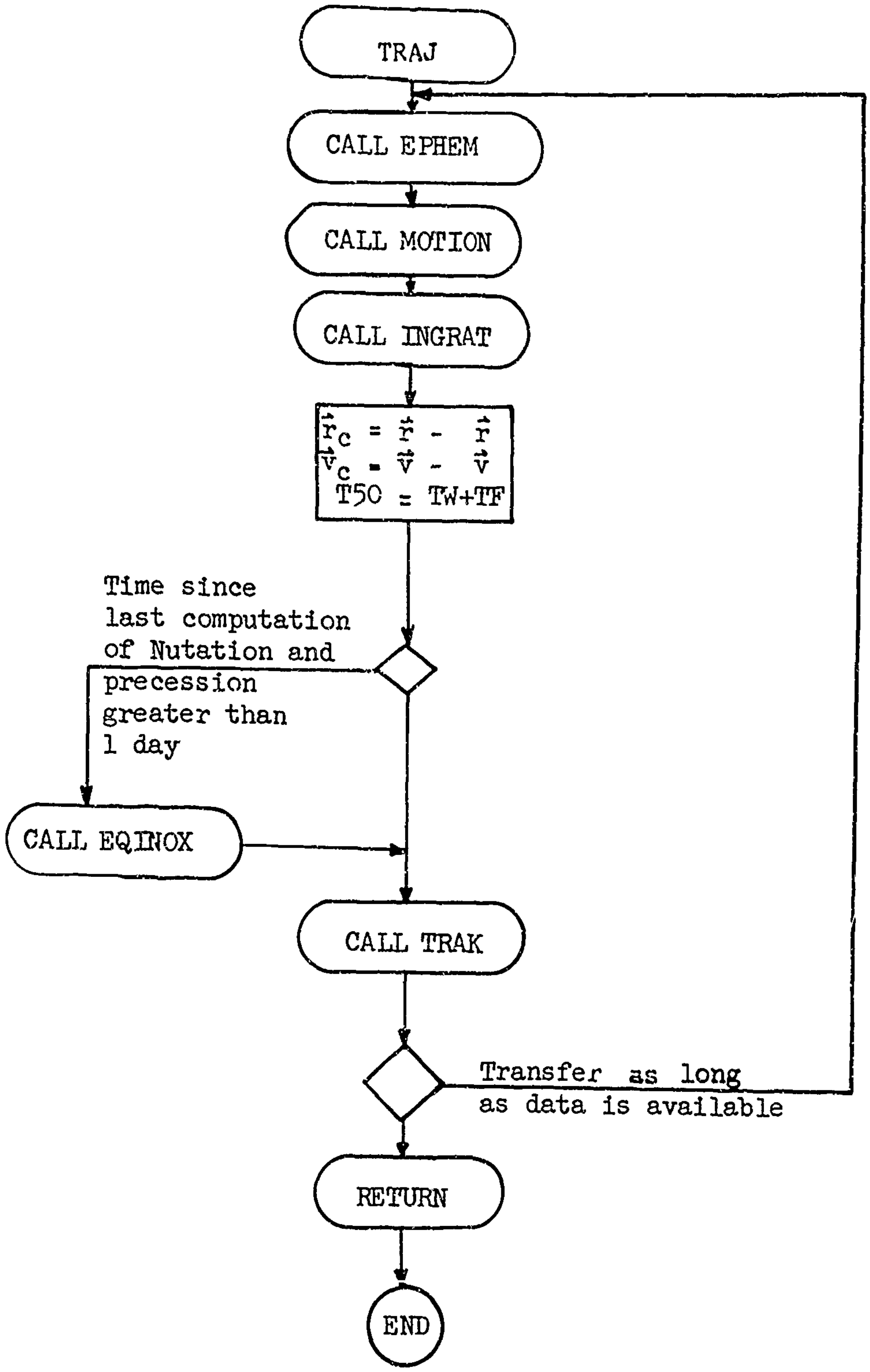
Discussion:

TRAJ does not in itself accomplish any portion of the required solution. Rather it is constructed for the purpose of sequencing the operations which are performed in generating the trajectory and checking the stations for visibility. In this sense, then, TRAJ is a direct extension of the program entitled MAIN and may appear unnecessary. However, for the purpose of checkout, it was deemed desirable to test the input and output functions of the program separately. Further, for the purpose of program appreciation and understanding it was deemed preferable to retain the checkout structure.

The first step taken upon entry into TRAJ is the computation of the solar and lunar position vectors (EPHEM) to aid in the computation of the gravitational and solar pressure contributions to the acceleration vector. This done, MOTION is entered for the purpose of computing the total acceleration vector and INGRAT is called for the purpose of estimating the position and velocity vectors at a time subsequent to the present epoch, (the step size for this integration is determined by the time interval between two successive data points and the magnitude of fifth difference of the acceleration vector as described in the discussion of the integration package). At this point, all vectors are known at the new epoch, time has been stepped, etc. and the conversion to the true equator of date from the mean equator of 1950.0 is defined. (For the sake of efficiency, this operation is performed only at one day intervals of time.) As a final step TRAK is called to determine if any of the tracking stations being employed observe the satellite and if there is data available at this epoch to process. (in the latter case, TRAK calls FILTER).

A transfer is constructed in TRAJ immediately after return from TRAK back to the computation of new solar and lunar position vectors. The cycle then repeats. Thus, on the surface there appears to be no means for the program to terminate. This conclusion is improper since at the time the last data point is processed, execution is terminated from within FILTER.

Computational Logic:



```
      SUBROUTINE TRAJ                                                   TRAJ0020
C                                                                       TRAJ0030
C     THIS ROUTINE SERVES AS THE DRIVER ROUTINE FOR THE EPHEMERIS,      TRAJ0040
C               MOTION, FOR INTEGRATION, AND FOR TRACKING .             TRAJ0050
C               ITS FUNCTION IS SIMPLY TO STEP THE VEHICLE ONE INCREMENT TRAJ0060
C               ALONG THE TRAJECTORY AND CHECK TO SEE IF THE VARIOUS    TRAJ0070
C               TRACKING STATIONS OF THE PROBLEM WERE VISIBLE .         TRAJ0080
C               ALL OUTPUT OF THE ROUTINE IS IN THE FRAME OF 1950.0     TRAJ0090
C     TW,TF = WHOLE AND FRACTIONAL DATE RELATIVE TO 1950.0              TRAJ0100
C     TIME  = TIME FROM LAST CONIC RECTIFICATION                        TRAJ0110
C     IRECT = VARIABLE USED TO SIGNAL CONIC RECTIFICATION               TRAJ0120
C     R,VVEC= RADIUS AND VELOCITY VECTORS  (1950.0)                     TRAJ0130
C     R,RD =THE POSITION AND VELOCITY RELATIVE TO THE ELLIPTIC REFER.   TRAJ0140
C     R,VCONIC= POSITION AND VELOCITY ON THE REFERENCE PATH             TRAJ0150
C                                                                       TRAJ0160
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * TRAJ0170
C                                                                       TRAJ0180
      DIMENSION  WRK(1)                                                 TRAJ0190
C                                                                       TRAJ0200
      COMMON  DATA                                                      TRAJ0210
C                                                                       TRAJ0220
      EQUIVALENCE           (DATA(391),WRK)                             TRAJ0230
C                                                                       TRAJ0240
      DIMENSION  RVEC(3)    ,VVEC(3)    ,R(3)       ,RD(3)      ,RCONIC(3)  TRAJ0250
     2,VCONIC(3)                                                        TRAJ0260
C                                                                       TRAJ0270
      EQUIVALENCE           (WRK( 34),TCONW ), (WRK( 35),TCONF )        TRAJ0280
     2                     ,(WRK( 44),RVEC  ), (WRK( 47),VVEC  )        TRAJ0290
     3                     ,(WRK( 50),TW    ), (WRK( 51),TF    )        TRAJ0300
     4                     ,(WRK( 52),R     ), (WRK( 55),RD    )        TRAJ0310
     5                     ,(WRK(104),TIME  )                           TRAJ0320
     6                     ,(WRK(117),RCONIC), (WRK(120),VCONIC)        TRAJ0330
C                                                                       TRAJ0340
      DATA ISTART,SAVE,III/1,0.,1/                                      TRAJ0350
C                                                                       TRAJ0360
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * TRAJ0370
```

```
C                                                                        TRAJ0380
C                                                                        TRAJ0390
C     COMPUTE ACCELERATION VECTOR * * * * * * * * * * * * * * * * * * * * TRAJ0400
    5 CALL EPHEM                                                         TRAJ0410    2
      CALL MOTION(ISTART,1)                                              TRAJ0420
C                                                                        TRAJ0430
C                                                                        TRAJ0440
C     STEP THE RELATIVE POSITION AND VELOCITY   * * * * * * * * * * * * TRAJ0450    4
      TIME = (( TW-TCONW ) + ( TF-TCONF ))*86400.                        TRAJ0460
      CALL INGRAT(ISTART)                                                TRAJ0470    6
      DO 10 I=1,3                                                        TRAJ0480
      RCONIC(I) = RVEC(I) - R(I)                                         TRAJ0490
   10 VCONIC(I) = VVEC(I) -RD(I)                                         TRAJ0500
C                                                                        TRAJ0510
C                                                                        TRAJ0520
C     CHECK FOR STATION VISIBILITY  * * * * * * * * * * * * * * * * * * TRAJ0530
      TFIFTY = TW + TF                                                   TRAJ0540
C     CALL EQINOX AT INTERVALS OF ONE DAY * * * * * * * * * * * * * *    TRAJ0545
      IF((TFIFTY-SAVE)-1. ) 30,30,40                                     TRAJ0550
   40 CALL EQINOX(TFIFTY)                                                TRAJ0560   22
      SAVE = TFIFTY                                                      TRAJ0570
   30 CALL TRAK(ISTART)                                                  TRAJ0580   25
      IF(III-1) 5,5,50                                                   TRAJ0590
   50 CONTINUE                                                           TRAJ0600
      RETURN                                                             TRAJ0600
      END                                                                TRAJ0610
```

SUBROUTINE TRAJ

COMMON VARIABLES

COMMON BLOCK // ORIGIN 00001 LENGTH 01000

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DATA	00000	R	WRK	00606	R	TCONW	00647	R
TCONF	00650	R	RVEC	00661	R	VVEC	00664	R
TW	00667	R	TF	00670	R	R	00671	R
RD	00674	R	TIME	00755	R	RCONIC	00772	R
VCONIC	00775	R						

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
ISTART	01001	I	TFIFTY	01002	R	SAVE	01003	R
III	01004	I						

ENTRY POINTS

TRAJ SECTION 4

SUBROUTINES CALLED

EPHEM	SECTION	5	MOTION	SECTION	6	INGRAT	SECTION	7
EQINOX	SECTION	8	TRAK	SECTION	9	SYSLOC	SECTION	10

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
5	1A	01014	10	14A	01045	30	24A	01070

```
    TRAJEC                                           STORAGE  MAP

40          21A        01062      50          28A        01100
 DECK LENGTH IN OCTAL IS     00117.
```

2.3.1.1 Conic Reference Motion Group

The trajectory for the space vehicle is formulated in the Encke Manner to assure numerical precision. That is, the difference in the acceleration vectors for a vehicle moving on the true trajectory and an imaginary vehicle which is moving along a conic trajectory (selected in such a manner that $\vec{r}_0$ and $\vec{v}_0$ are equal for the two trajectories) is evaluated at every point. This differential acceleration is then integrated to obtain the differential position and velocity vectors and the true motion constructed by adding the reference position and velocities to the corrections.

This group of routines is designed to provide the reference $\vec{r}$ and $\vec{v}$ as functions of the time utilizing a deterministic set of variables to assure numerical significance in all computations. The group is completely self contained save for several math functions which are provided elsewhere in the program and for the gravitational constant which is acquired from common.

Subroutine CONIC

Purpose: CONIC provides the conic reference trajectory to be utilized in conjunction with an Encke integration being performed in other portions of the program. The variables are those recommended by Dr. S. Herrick and are employed to avoid ambiguities in the solution as the eccentricity approaches zero or one and/or as the inclination of the reference orbit plane approaches zero.

Deck Name: CON

Calling Sequence: CALL CØNIC (RØ, SØ, TIME, X, DTIME, R, V)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	RO SO	$\vec{r}_{co}$ $\vec{v}_{co}$	3 3	Arg Arg	The position and velocity vectors at an arbitrary epoch in cartesian coordinates (Km, Km/sec.)
I	TIME	t	1	Arg	The time in seconds from the epoch of $\vec{r}_o$, $\vec{v}_o$ to the epoch at which $\vec{r}$ and $\vec{v}$ are desired.
I/O	X	X	1	Arg	The eccentric anomaly variables $(E - E_o)$ defining the position at the desired time. The quantity is saved to aid in future iterative solutions for $X = X(\ \ t)$
I	DTIME	t	1	Arg	The incremental time in seconds since the previous solution yielded the value of X being fed back into CONIC

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
O	R V	$\vec{r}$ $\vec{v}$	3 3	Arg	The radius and velocity vectors at time t (in the same coordinate frame as $\vec{r}_o$, $\vec{v}_o$) expressed in Km and Km/sec.
I	GM	μ	1	CON(6)	The gravitational constant for the central body (Km ^{3}/sec^2)

Subroutines Required: SEARCH (numerical-analytic search for the position variable satisfying the time constraint)

Functions Required:

- DOT (vector dot product)
- AMAG (vector magnitude)
- COSH (hyperbolic cosine)
- SINH (hyperbolic sine)
- COS (cosine)
- SIN (sine)
- SQRT (square root)

Approximate Deck Length: 420 (octal)

Formulation:

The equations of motion for the central force problem are

$$\ddot{\vec{r}} = -\frac{\mu}{r^3}\vec{r} \tag{1}$$

Thus, the rate of change of the angular momentum vector is

$$\frac{d}{dt}(\vec{h}) = \vec{r} \times \vec{F}/m = \vec{r} \times \left(-\frac{\mu}{r^3}\right)\vec{r} = 0 \tag{2}$$

This equation states that the angular momentum (thus the plane of motion) of the resultant conic is constant and leads directly to the fact that any vector in the plane can be constructed by linear combination of any two vectors selected. i.e.,

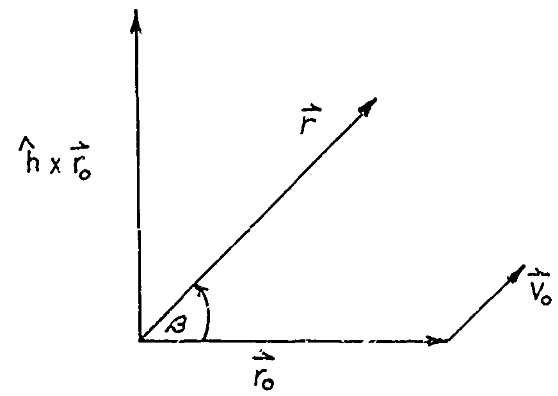
$$\vec{r} = f\vec{r}_0 + g\vec{v}_0 \tag{3}$$

Further, since $\vec{r}_0$ and $\vec{v}_0$ are constants and since $\vec{v} = d/dt\ (\vec{r})$

$$\vec{v} = \dot{f}\,\vec{r}_0 + \dot{g}\,\vec{v}_0 \tag{4}$$

This representation of $\vec{r}$ and $\vec{v}$ is completely definitive for <u>all</u> conic trajectories and exhibits none of the ambiguities or indeterminances encountered with the solution employing the classic set of elements (a, e, i, ω, Ω , M_0). For this reason, equations 3 and 4 will be employed in this program and attention will be turned to the evaluation of the quantities f, g, $\dot{f}$ and $\dot{g}$

Consider the following sketch which illustrates motion in the orbit plane



$(\hat{h} = \vec{h}/|\vec{h}|)$

$$\vec{r} = \frac{r}{r_0} \cos\beta\, \vec{r}_0 + \frac{r}{r_0} \sin\beta\, \hat{h} \times \vec{r}_0$$

$$= \frac{r}{r_0} \cos\beta\, \vec{r}_0 + \frac{r}{h r_0} \sin\beta \left[\vec{V}_0\, r_0^2 - \vec{r}_0 \left(\vec{r}_0 \cdot \vec{V}_0\right)\right]$$

$$= \frac{r}{r_0} \left(\cos\beta - \frac{\vec{r}_0 \cdot \vec{V}_0}{h} \sin\beta\right) \vec{r}_0 + \frac{r r_0}{h} \sin\beta\, \vec{V}_0$$

Now since equations 3 and 5 represent the same vector

$$f = \frac{r}{r_0} \left(\cos\beta - \frac{\vec{r}_0 \cdot \vec{V}_0}{h} \sin\beta \right) \tag{6}$$

$$g = \frac{r r_0}{h} \sin\beta \tag{7}$$

where $h = \sqrt{\mu p} = \sqrt{\mu\, a\,(1 - e^2)}$

The basic problem at this point now reduces itself to the evaluation of the angle β in terms of variables which can be related to time through the dynamics of the motion. This task requires that equation 1 be integrated and that the solution shown to be conic. However, since this process has been accomplished in such a large number of references (e.g., Reference 1), the solution is assumed to be known and the angle β is recognized to be the difference in the true anomalies at positions corresponding to $\vec{r}_0$ and $\vec{r}$.

Under these observations the trigonometric functions $\sin\beta$ and $\cos\beta$ may be reduced to functions of the eccentric anomaly (selected due to the fact that Kepler's equation will be utilized to obtain f, g, $\dot{f}$ and $\dot{g}$ as function of time) by substituting the following identities:

$$\sin\theta = \frac{a}{r} \sqrt{1 - e^2} \sin E \tag{8}$$

$$\cos\theta = \frac{a}{r} \left[\cos E - e\right] \tag{9}$$

Thus

$$\sin\beta = \sin(\theta - \theta_0) = \sin\theta \cos\theta_0 - \sin\theta_0 \cos\theta$$

$$= \frac{a^2}{r r_0} \sqrt{1-e^2}\left[SE(CE_0 - e) - SE_0(CE - e)\right]$$

$$= \frac{a^2}{r r_0} \sqrt{1-e^2}\left[(SECE_0 - SE_0\,CE) - e(SE - SE_0)\right]$$

$$= \frac{a^2}{r r_0} \sqrt{1-e^2}\left[sX - e(SE - SE_0)\right] \tag{10}$$

where: the notation $s\alpha = \sin\alpha$; $c\alpha = \cos\alpha$ and $X = E - E_0$

but $SE = \sin(E_0 + X)$

$= sXcE_0 + cX\,SE_0$

so that

$$\sin\beta = \frac{a^2}{r r_0} \sqrt{1-e^2}\left[sX(1 - eCE_0) + e\,SE_0(-cX + 1)\right]$$

$$= \frac{a^2}{r r_0} \sqrt{1-e^2}\left[\frac{r_0}{a}\, sX + \frac{\vec{r}_0 \cdot \vec{V}_0}{\sqrt{\mu a}} (1 - cX)\right] \tag{11}$$

similarly

$$\cos\beta = \cos\theta \cos\theta_0 + \sin\theta \sin\theta_0$$

$$= \frac{a^2}{r r_0}\left[(CE - e)(CE_0 - e) + (1 - e^2)\, SE\, SE_0\right]$$

$$= \frac{a^2}{r r_0}\left[(CECE_0 + SESE_0) + e^2(1 - SE\,SE_0) - e(CE + CE_0)\right]$$

$$= \frac{a^2}{r r_0}\left\{ cX + e^2\left[1 - (sXCE_0 + cXSE_0)\, SE_0\right] - e(cXCE_0 - sXSE_0 + CE_0)\right\}$$

$$= \frac{a^2}{rr_0}\Big[cx + e^2 - e^2\, sx\, cE_0\, sE_0 - e^2\, cx\, sE_0^{\,2}$$

$$- ec\, x\, cE_0 + e\, sx\, sE_0 - e\, cE_0\Big] \tag{12}$$

which can be grouped and reduced as follows

$$\cos\beta = \frac{a^2}{rr_0}\Big[\left(cx - ecx\,cE_0\right) + e^2 sE_0^2\, cx +$$

$$sx\left(-e^2 cE_0\, sE_0 + e\, sE_0\right) + \left(e^2\right) - \left(ecE_0\right)\Big]$$

$$= \frac{a^2}{rr_0}\Big[\frac{r_0}{a}\, cx - \frac{(\vec{r}_0\cdot\vec{V}_0)^2}{\mu a}\, cx$$

$$+ sx\left(\frac{r_0}{a}\ \frac{\vec{r}_0\cdot\vec{V}_0}{\sqrt{\mu a}}\right) + \left(1 - \frac{p}{a}\right) - \left(1 - \frac{r_0}{a}\right)\Big]$$

$$= \frac{a^2}{rr_0}\Big[\ \frac{r_0}{a}\left(cx + 1\right) - \frac{(\vec{r}_0\cdot\vec{V}_0)^2}{\mu a}\, cx + sx\left(\frac{r_0}{a}\ \frac{\vec{r}_0\cdot\vec{V}_0}{\sqrt{\mu a}}\right)$$

$$- \frac{r_0^2\, V_0^2}{\mu a}\left(1 - \frac{(\vec{r}_0\cdot\vec{V}_0)^2}{r_0^2\, V_0^2}\right)\Big] \tag{13}$$

$$\cos\beta = \frac{a^2}{rr_0}\Big[\frac{r_0}{a}\left(cx + 1\right) + \frac{(\vec{r}_0\cdot\vec{V}_0)^2}{\mu a}\left(1 - cx\right) + sx\left(\frac{r_0}{a}\ \frac{\vec{r}_0\cdot\vec{V}_0}{\sqrt{\mu a}}\right)$$

$$- \frac{r_0^2\, V_0^2}{\mu a}\Big]$$

$$= \frac{a^2}{rr_0}\Big[-\frac{r_0}{a}\left(1 - cx\right) + \left(\frac{\vec{r}_0\cdot\vec{V}_0}{\mu a}\right)^2\left(1 - cx\right) + sx\left(\frac{r_0}{a}\ \frac{\vec{r}_0\cdot\vec{V}_0}{\sqrt{\mu a}}\right)$$

$$- \frac{r_0}{a}\left(\frac{r_0 V_0^2}{\mu} - 2\right)\Big]$$

$$= \frac{a^2}{rr_0}\ \Big\{\left(1 - cx\right)\Big[\frac{(\vec{r}_0\cdot\vec{V}_0)^2}{\mu a} - \frac{r_0}{a}\Big] + sx\left(\frac{r_0}{a}\ \frac{\vec{r}_0\cdot\vec{V}_0}{\sqrt{\mu a}}\right)$$

$$+ \left(\frac{r_0}{a}\right)^2\Big\} \tag{14}$$

Finally, Kepler's equation in terms of the time of transit from $\vec{r}_0$ to $\vec{r}$ can be obtained as

$$\sqrt{\frac{\mu}{a^3}}\,(t - t_0) = (E - e\,sE) - (E_0 - e\,sE_0)$$

$$= X - e\left(sE - sE_0\right)$$

$$= X - e\left(sX\,cE_0 + cX\,sE_0 - sE_0\right)$$

$$= X + sX\left(-e\,cE_0\right) - e\,sE_0\left(1 - cX\right)$$

$$= X - \left(1 - \frac{r}{a}\right) sX - \frac{\vec{r}_0 \cdot \vec{V}_0}{\sqrt{\mu a}}\left(1 - cX\right)$$

$$= (X - sX) + \frac{r_0}{a}\,sX - \frac{\vec{r}_0 \cdot \vec{V}_0}{\sqrt{\mu a}}\left(1 - cX\right) \qquad (15)$$

and the quantities f and g determined as follows

$$f = \frac{r}{r_0}\left(\cos\beta - \frac{\vec{r}_0 \cdot \vec{V}_0}{\sqrt{\mu p}}\sin\beta\right)$$

$$= \left(\frac{a}{r_0}\right)^2 \left\{ (1 - cX)\left[\frac{(\vec{r}_0 \cdot \vec{V}_0)^2}{\mu a} - \frac{r_0}{a}\right] + sX\left(\frac{r_0}{a}\,\frac{\vec{r}_0 \cdot \vec{V}_0}{\sqrt{\mu a}}\right.\right.$$

$$\left.\left. + \left(\frac{r_0}{a}\right)^2\right\} + \left(\frac{a}{r_0}\right)^2 \frac{\sqrt{1-e^2}\;\vec{r}_0 \cdot \vec{V}_0}{\sqrt{\mu a(1-e^2)}}\left\{\frac{r_0}{a}\,sX\right.$$

$$\left. + \frac{\vec{r}_0 \cdot \vec{V}_0}{\sqrt{\mu a}}\left(1 - cX\right)\right\}$$

or $$f = 1 - \frac{a}{r_0}(1 - cX). \tag{16}$$

similarly

$$g = \frac{r r_0}{\sqrt{\mu a (1-e^2)}} \left\{ \frac{a^2}{r r_0} \sqrt{1-e^2} \left[\frac{r_0}{a} SX + \frac{\vec{r}_0 \cdot \vec{V}_0}{\sqrt{\mu a}} (1 - cX) \right] \right\}$$

$$= \sqrt{\frac{a^3}{\mu}} \left[\frac{r_0}{a} SX + \frac{\vec{r}_0 \cdot \vec{V}_0}{\mu a} (1 - cX) \right]$$

which may be simplified by observing the form of Kepler's equation to yield

$$g = \sqrt{\frac{a^3}{\mu}} \left[\frac{r_0}{a} SX + \sqrt{\frac{\mu}{a^3}} (t - t_0) - (X - SX) - \frac{r_0}{a} SX \right]$$

$$= (t - t_0) - \sqrt{\frac{a^3}{\mu}} (X - SX) \tag{17}$$

The coefficients $\dot{f}$ and $\dot{g}$ can now be determined by differentiating Kepler's equation as follows:

$$\frac{d}{dt}\left[\sqrt{\frac{\mu}{a^3}} (t - t_0)\right] = \frac{d}{dt}\left[(E - e \sin E) - (E_0 - e \sin E_0)\right]$$

$$\sqrt{\frac{\mu}{a^3}} = \dot{E}(1 - e \cos E)$$

$$= \dot{E} \frac{r}{a}$$

or $$\dot{E} = \dot{X} = \frac{1}{r}\sqrt{\frac{\mu}{a}} \tag{18}$$

and by returning to equation 3 to note that

$$\dot{f} = \frac{d}{dt}(f)$$

$$= \frac{a}{r_o}\frac{d}{dt}(CX) = -\frac{a}{r_o} SX \ \dot{X}$$

$$= -\frac{\sqrt{a\mu}}{r r_o} SX \tag{19}$$

and

$$\dot{g} = \frac{d}{dt}(g)$$

$$= 1 - \sqrt{\frac{a^3}{\mu}}(1 - CX)\left(\frac{1}{r}\sqrt{\frac{\mu}{a}}\right)$$

$$= 1 - \frac{a}{r}(1 - CX) \tag{20}$$

Equations 3, 4, 15, 16, 17, 19, and 20 now provide the complete description of the conic motion problem in a completely deterministic set of variables. These equations are utilized as follows to define $\vec{r}$ and $\vec{v}$ as functions of $t - t_o$:

1) solve 15 for X (iterative)

2) solve 16, 17, 19, and 20 for f, g, $\dot{f}$ and $\dot{g}$

3) solve 3 and 4 for $\vec{r}$ and $\vec{v}$

At this point Dr. S. Herrick's notation in "Universal" variables (Reference 2) can be adopted by defining the following quantities

$$\hat{S} = \sqrt{a} \ SX$$

$$\hat{C} = a(1 - CX)$$

$$\hat{V} = a^{3/2}(X - SX)$$

$$\tau = \sqrt{\mu}\,(t - t_o)$$

$$\vec{V}' = \vec{V}/\sqrt{\mu}$$

$$\vec{r} = f'\vec{r}_0 + g'\vec{V}'$$
$$\vec{V}' = \dot{f}'\vec{r}_0 + \dot{g}'\vec{V}'$$

and f', g', $\dot{f}'$, and $\dot{g}'$ can now be expressed in the form in which they will be coded.

$$f' = 1 - \frac{\hat{c}}{r_0}$$

$$g' = \sqrt{\mu}\, g = \tau - \hat{U}$$

$$\dot{f}' = \frac{\dot{f}}{\sqrt{\mu}} = \frac{-\hat{s}}{r\, r_0}$$

$$\dot{g}' = 1 - \frac{\hat{c}}{r}$$

It is noted that at every step in the development of these equations the implicit assumption of elliptic motion has been made (see equation 8). However, since for hyperbolic motion

$$a_h = \text{negative}$$

$$F = i\,E$$

$$F - F_0 = \Psi = i\,X$$

only the following modifications need be made to provide for a hyperbolic reference orbit capability

$$\hat{s} = -\sqrt{-a_h}\,\sinh \Psi$$

$$\hat{c} = a_h(1 - \cosh \Psi)$$

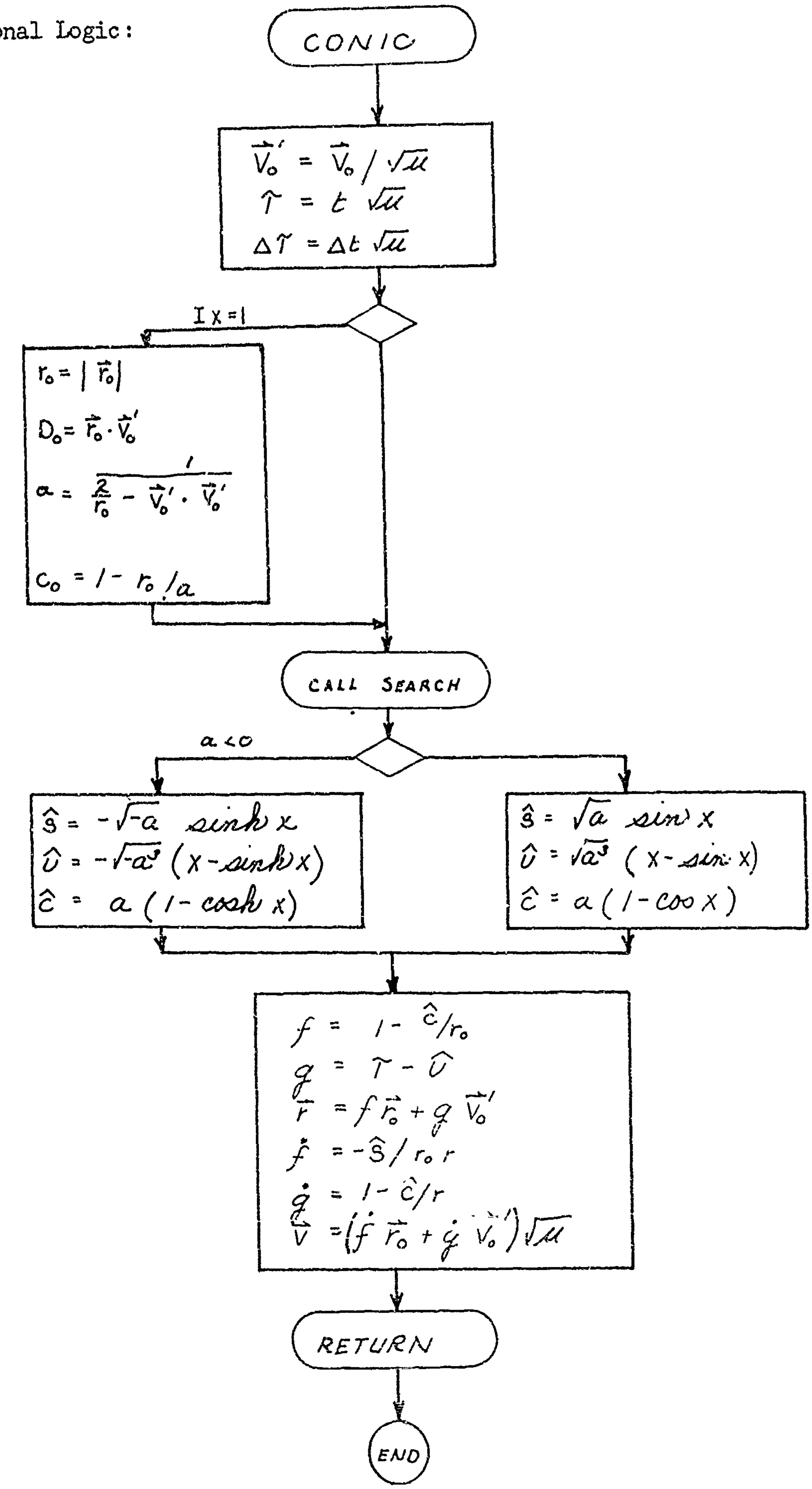
$$\hat{U} = -\sqrt{-a_h^3}\,(\Psi - \sinh \Psi)$$

Thus, this generalization will be included.

1. Townsend, G. E., "Orbital Mechanics" Chapter 3 in The Orbital Flight Handbook, NASA SP-33 Part 1, 1964.

2. Herrick, S., "Universal Variables" published by the author, University of California (L.A.) November 2, 1964.

Computational Logic:



```
      SUBROUTINE CONIC(RO,SO,TIME,X,DTIME,R,V)                          CONIC020
C                                                                       CONIC030
C                                                                       CONIC040
C     THIS ROUTINE IS THE DRIVER ROUTINE FOR SEVERAL SUBPROGRAMS        CONIC050
C             DESIGNED TO COMPUTE THE POSITION AND VELOCITY VECTORS     CONIC060
C             ON CONIC SECTIONS AS FUNCTIONS OF THE TIME. THE VARIABLESCONIC070
C             OF THIS ANALYSIS ARE HERRICKS UNIVERSAL VARIABLES .       CONIC080
C     RO = INITIAL POSITION VECTOR                                      CONIC090
C     SO = INITIAL VELOCITY VECTOR (DUMMY NAME DUE TO UNITS OF V        CONIC100
C             IN UNIVERSAL FORMULATION )                                CONIC110
C     IX = FIXED POINT NUMBER TO DENOTE THE EXISTANCE OF THE ELEMENTS   CONIC120
C     ELEM=AN ARRAY OF ELLIPTIC ELEMENTS ( RO,DO,AO,CO )                CONIC130
C     TIME= TRUE TIME IN UNITS CONSISTENT WITH GM RELATIVE TO RO,VO     CONIC140
C     X  = THE MEASURE OF POSITION (ELLIPTIC OR HYPERBOLIC ) RELATIVE   CONIC150
C             TO THE INITIAL POSITION ** ENTER WITH PREVIOUS ESTIMATE   CONIC160
C             TO ASSIST IN SEARCH AS FUNCTION OF TIME ,ESTIMATE UPDATEDCONIC170
C     DTIME=CHANGE IN TIME VARIABLE RELATIVE TO PREVIOUS ESTIMATE       CONIC180
C     R  = POSITION VECTOR AT TAU                                       CONIC190
C     V  = VELOCITY VECTOR AT TAU                                       CONIC200
C                                                                       CONIC210
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   CONIC220
C                                                                       CONIC230
      DIMENSION  CON(1)  ,WRK(1)                                        CONIC240
C                                                                       CONIC250
      COMMON DATA                                                       CONIC260
C                                                                       CONIC270
      EQUIVALENCE           (DATA(  1),CON  ), (DATA(391),WRK)          CONIC280
C                                                                       CONIC290
      DIMENSION RO(3),VO(3),ELEM(4),R(3),V(3),SO(3)                     CONIC300
C                                                                       CONIC310
      EQUIVALENCE           (CON(  6),GM   ), (WRK(130),ELEM  )         CONIC320
C                                                                       CONIC330
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   CONIC340
C                                                                       CONIC350
      SQ = SQRT(GM)                                                     CONIC360
      DO 60 I=1,3                                                       CONIC370
```

```
     CONC         -  EFN    SOURCE STATEMENT  -  IFN(S)  -

 60 VO(I) = SO(I) / SQ                                      CONIC380
    TAU = TIME * SQ                                         CONIC390
    DTAU = DTIME * SQ                                       CONIC400
    ELEM(1) = AMAG(RO)                                      CONIC410    11
    ELEM(2) = DOT(RO,VO)                                    CONIC420    12
    ELEM(3) = 1./( 2./ELEM(1) - DOT(VO,VO) )                CONIC430    13
    ELEM(4) = 1. - ELEM(1)/ELEM(3)                          CONIC440
    CALL SEARCH(TAU,DTAU,X)                                 CONIC450    14
    OR=ELEM(1)                                              CONIC460
    A=ELEM(3)                                               CONIC470
    IF(A) 10,20,20                                          CONIC480
 10 SHAT =-SQRT(-A)* SINH(X)                                CONIC490    18    19
    UHAT =-SQRT(-A*A*A) *(X-SINH(X))                        CONIC500    20    21
    CHAT = A*(1.-COSH(X))                                   CONIC510    22
    GO TO 30                                                CONIC520
 20 UHAT = SQRT( A*A*A) *(X-SIN(X) )                        CONIC530    25    26
    CHAT = A*(1.-COS(X) )                                   CONIC540    27
    SHAT = SQRT( A)* SIN(X)                                 CONIC550    28    29
 30 F = 1.-CHAT/OR                                          CONIC560
    G = TAU -UHAT                                           CONIC570
    DO 40 I=1,3                                             CONIC580
 40 R(I) = F *RO(I) + G * VO(I)                             CONIC590
    R1=AMAG(R)                                              CONIC600    40
    FDOT= -SHAT/(OR*R1)                                     CONIC610
    GDOT=1. -CHAT/R1                                        CONIC620
    DO 50 I=1,3                                             CONIC630
 50 V(I) =(FDOT*RO(I) + GDOT*VO(I))*SQ                      CONIC640
    RETURN                                                  CONIC650
    END                                                     CONIC660
```

****
CONC

STORAGE MAP 01/20/86 PAGE 49

SUBROUTINE CONIC

COMMON VARIABLES

COMMON BLOCK // ORIGIN 00001 LENGTH 01013

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DATA	00000	R	DATA	00000	R	CON	00000	R
WRK	00606	R	GM	00005	R	ELEM	01007	R

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
VO	01014	R						

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
SQ	01017	R	TAU	01020	R	DTAU	01021	R
OR	01022	R	A	01023	R	SHAT	01024	R
UHAT	01025	R	CHAT	01026	R	F	01027	R
G	01030	R	R1	01031	R	FDOT	01032	R
GDOT	01033	R						

ENTRY POINTS

CONIC	SECTION	4

SUBROUTINES CALLED

SQRT	SECTION	5	AMAG	SECTION	6	DOT	SECTION	7
SEARCH	SECTION	8	SINH	SECTION	9	COSH	SECTION	10

```
 ****                                                                      01/20/86              PAGE  50
     CCNC                                         STORAGE  MAP

   SIN         SECTION      11          COS          SECTION      12          SYSLOC       SECTION      13

                                                EFN    IFN    CORRESPONDENCE

EFN           IFN          LOCATION    EFN          IFN          LOCATION    EFN          IFN          LOCATION
60             6A           01052      10            17A          01141      20            24A          01220
30            30A           01270      40            34A          01302      50            44A          01337

  DECK LENGTH IN OCTAL IS      00420.
```

2.3.1.1.1 Subroutine SEARCH

Purpose: SEARCH is the iterative logic utilized to solve Kepler's equation for the position variable X (see CONIC) as a function of the time elapsed since the epoch of r_o, v_o.

Deck Name: LOOK

Calling Sequence: CALL SEARCH (TAU, DTAU, X)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	TAU	τ	1	Arg	Universal time variable defined by $\tau = \sqrt{u}\, t$
I	DTAU	$\Delta\tau$	1	Arg	the change in tau since the solution was last iterated. (Tau for first solution)
I	ELEM	T_o D_o a C_o	1 1	WRK(130)	the array of constants describing the nature of the motion
O	X	X	1	Arg	the eccentric (hyperbolic) anomaly variable defining position on a conic section as a function of $t - t_o$

Subroutines Required: None

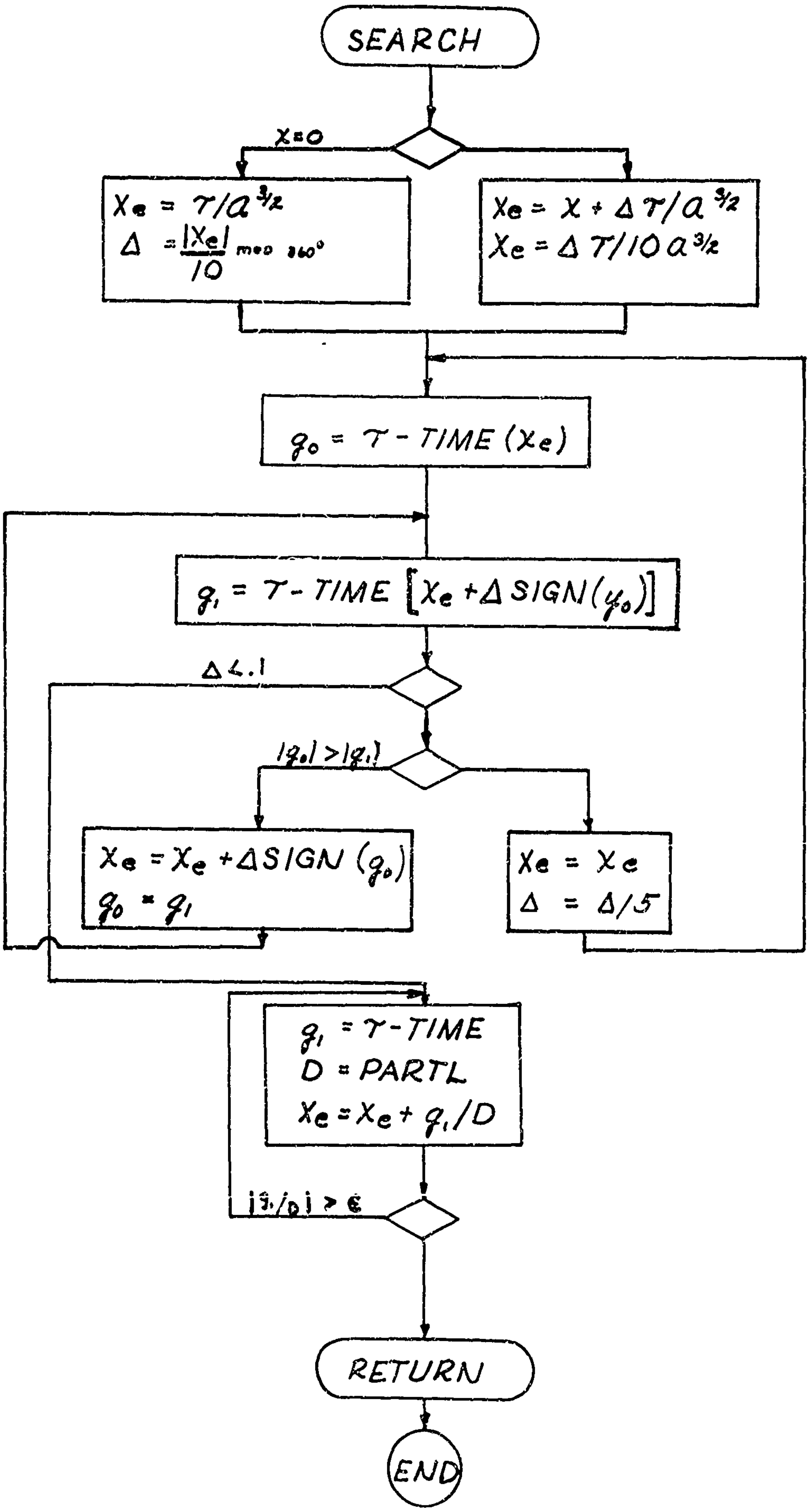
Function Required: PARTL (partial derivative of Keplers equation with respect to X)

Function Required (continued)	TIME	(Time relative to the epoch of r_o, v_o)
	SQRT	(square root)
	SIGN	(function for attaching the sign of one variable to another)
	ABS	(absolute value)
Approximate Deck Length:	190	(decimal)

Discussion:

SEARCH solves a monotonic transcendental equation of one degree of freedon in a two stage iteration process. First, a guess of the solution is made and the function evaluated at the guess and at the guess plus a fixed increment. The error in the functions are then computed and the value of guess which produced the smallest error in an absolute sense is saved. The process then repeats itself until the desired solution is approached. At this time, the step size is reduced and the search continued in stages until the error in the desired root is small enough to assure that a Newton type iteration will converge to the answer. At this time a variable step size is computed based on the error in the function and the slope of the transcendental equation at this argument.

SEARCH may, thus, be constructed as a general purpose routine for functions of this type with provision for incorporating a means of generating the initial guess, the transcendental function being solved, and an analytic partial of the function with respect to its argument. This approach has been taken in the routine being discussed with minor revision for the sake of brevity and with only the specific application in mind (that of solving Kepler's equation of the anomaly variable as a function of time).


SEARCH
X=0
Xe = T/a^3/2
Δ = |Xe|/10 mod 360°
Xe = X + ΔT/a^3/2
Xe = ΔT/10a^3/2
q0 = T - TIME (Xe)
q1 = T - TIME [Xe + Δ SIGN(q0)]
Δ < .1
|q0| > |q1|
Xe = Xe + Δ SIGN (q0)
q0 = q1
Xe = Xe
Δ = Δ/5
q1 = T - TIME
D = PARTL
Xe = Xe + q1/D
|q1/D| > ε
RETURN
END

```
      SUBROUTINE SEARCH(TAU,DTAU,X)                                   LOOK0020
C                                                                     LOOK0030
C     THIS ROUTINE SEARCHES FOR THE ROOT OF A MONOTONICALLY INCREASING  LOOK0040
C           TRANSCENDENTAL EQUATION ** ITS OPERATION IS INDEPENDENT   LOOK0050
C           OF THE NATURE OF THE FUNCTION ** CONVERGENCE IS OBTAINED  LOOK0060
C           IN TWO STAGES ,SEARCH AND NEWTON ITERATION                LOOK0070
C                                                                     LOOK0080
      DIMENSION  WRK(1)                                               LOOK0090
C                                                                     LOOK0100
      COMMON DATA                                                     LOOK0110
C                                                                     LOOK0120
      EQUIVALENCE          (DATA(391),WRK   )                         LOOK0130
C                                                                     LOOK0140
      DIMENSION ELEM(4)                                               LOOK0150
C                                                                     LOOK0160
      EQUIVALENCE          (WRK(130),ELEM   )                         LOOK0170
C                                                                     LOOK0180
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * LOOK0190
C                                                                     LOOK0200
      A32= SQRT(ABS(ELEM(3))**3)                                      LOOK0210    2
      IX=1                                                            LOOK0220
      AA=1.                                                           LOOK0230
      IF(X) 1,2,1                                                     LOOK0240
    2 EST = TAU/A32                                                   LOOK0250
      TWOPI = 2.0*3.1415926                                           LOOK0252
      DEL = ABS(EST)                                                  LOOK0254
      ESTMOD = AMOD(DEL,TWOPI)                                        LOOK0256
      DEL = ESTMOD/10.0                                               LOOK0258
      GO TO 3                                                         LOOK0270
    1 EST = X + DTAU/A32                                              LOOK0280
      DEL = DTAU/(10.*A32)                                            LOOK0290
    3 G0 = TAU - TIME(EST)                                            LOOK0300    9
    4 G1 = TAU - TIME(EST + DEL*SIGN(AA,G0))                          LOOK0310   11
      AA=1.                                                           LOOK0320
   13 IF(ABS(DEL)-.1) 10,10,11                                        LOOK0330
   11 IF( ABS(G1)-ABS(G0) ) 12,10,10                                  LOOK0340
```

```
****                                                                    01/21/86           PAGE  17
      LØØK        -  EFN   SØURCE STATEMENT  -  IFN(S)  -

  12 FST = FST + DEL* SIGN(AA,GØ)                                       LØØK0350
     AA=1.                                                              LØØK0360
     IX=2                                                               LØØK0370
     GØ = G1                                                            LØØK0380
     GØ TØ 4                                                            LØØK0390
  19 IF(IX-1) 21,21,22                                                  LØØK0400
  22 EST = FST                                                          LØØK0410
  21 DEL = DEL/ 5.                                                      LØØK0420
     GØ TØ 3                                                            LØØK0430
  10 G1 = TAU-TIME(FST)                                                 LØØK0440     25
     DERIV = PARTL(EST)                                                 LØØK0450     26
     STEP = G1/DERIV                                                    LØØK0460
     EST = EST + STEP                                                   LØØK0470
     IX=IX+1                                                            LØØK0480
     IF(IX-100) 23,23,20                                                LØØK0490
  23 IF(ABS(STEP)-.0000001)20,10,10                                     LØØK0500
  20 X = FST                                                            LØØK0510
     RETURN                                                             LØØK0520
     END                                                                LØØK0530
```

LOOK

STORAGE MAP

SUBROUTINE SEARCH

COMMON VARIABLES

COMMON BLOCK // ORIGIN 00001 LENGTH 01013

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DATA	00000	R	WRK	00606	R	ELFM	01007	R

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
A32	01014	R	IX	01015	I	AA	01016	R
EST	01017	R	TWOPI	01020	R	DEL	01021	R
ESTMOD	01022	R	G0	01023	R	G1	01024	R
DERIV	01025	R	STEP	01026	R			

ENTRY POINTS

SEARCH SECTION 4

SUBROUTINES CALLED

SORT	SECTION	5	TIME	SECTION	6	PARTL	SECTION	7
CC.1	SECTION	8	CC.2	SECTION	9	CC.3	SECTION	10
CC.4	SECTION	11	SYSLOC	SECTION	12			

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1	7A	01122	2	5A	01071	3	8A	01135
4	10A	01144	13	12A	01164	10	24A	01232

```
****                                                          01/21/86              PAGE  29
        LOOK                              STORAGE  MAP

11         14A        01171      12          16A       01202      19          19A       01220
21         22A        01226      22          21A       01224      23          29A       01263
20         31A        01270
 DECK LENGTH IN OCTAL IS      00310.
```

2.3.1.1.2 Function TIME

Purpose: TIME is the function which computes the difference in the epochs at two points on an arbitrary conic section.

Deck Name: TIME

Calling Sequence: TIME (X)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	X	X	1	Arg	the eccentric or hyperbolic anomaly variable defining position relative to that of an initial epoch
I	ELEM	1. D_o a C_o	1 1 1 1	WRK(130)	array of constants used to describe the conic section.
O	TIME		1	-	the normalized time variable τ $(t - t_o)$

Subroutines Required: None

Functions Required:

COS (cosine)
SIN (sine)
COSH (hyperbolic cosines)
SINH (hyperbolic sine)
SQRT (square root)

Approximate Deck Length: 150 (decimal)

```
      TIM         -  EFN    SOURCE STATEMENT  -  IFN(S)  -

      FUNCTION TIME(X)                                          TIME0020
C                                                               TIME0C30
      DIMENSION  WRK(1)                                         TIME0040
C                                                               TIME0050
      COMMON DATA                                               TIME0060
C                                                               TIME0070
      EQUIVALENCE          (DATA(391),WRK   )                   TIME0080
C                                                               TIME0090
      DIMENSION ELEM(4)                                         TIME0100
C                                                               TIME0110
      EQUIVALENCE          (WRK(130),ELEM   )                   TIME0120
C                                                               TIME0130
      AA=1.                                                     TIME0140
      IF(ELEM(3)) 1,2,2                                         TIME0150
    1 UHAT=SQRT(-ELEM(3)**3)*(X-SINH(X))*(-1.)                  TIME0160   4     5
      CHAT=ELEM(3)*(1.-COSH(X))                                 TIME0170   6
      GO TO 30                                                  TIME0180
    2 UHAT=SQRT(ELEM(3)**3)*(X-SIN(X))                          TIME0190   9     10
      CHAT=ELEM(3)*(1.-COS(X))                                  TIME0200   11
   30 XHAT=SQRT(ABS(ELEM(3)))*X*SIGN(AA,ELEM(3))                TIME0210   13
      TIME=XHAT*ELEM(1)+CHAT*ELEM(2) +UHAT*ELEM(4)              TIME0220
      RETURN                                                    TIME0230
      END                                                       TIME0240
```

STORAGE MAP

FUNCTION TIME TYPE R

COMMON VARIABLES

COMMON BLOCK // ORIGIN CCCC1 LENGTH C1C13

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DATA	0C000	R	WRK	0C606	R	ELEM	C1007	R

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
F.000C	C1014	R	AA	01015	R	UHAT	C1016	R
CHAT	C1C17	R	XHAT	01020	R			

ENTRY POINTS

TIME SECTION 4

SUBROUTINES CALLED

SQRT	SECTION	5	SINH	SECTION	6	COSH	SECTION	7
SIN	SECTION	8	COS	SECTION	9	SYSLOC	SECTION	1C

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1	3A	C1C37	2	8A	C11C1	3C	12A	C1135

DECK LENGTH IN OCTAL IS CC204.

2.3.1.1.3 Function PARTL

Purpose: PARTL provides the derivative of Kepler's equation with respect to the anomaly variable (X).

Deck Name: PART

Calling Sequence: PARTL (X)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	X	X	1	Arg	the eccentric or hyperbolic anomaly variable defining position relative to that of an arbitrary initial epoch
I	ELEM	r_o D_o a	1 1 1	WRK(130)	array of constants used to describe the conic section
O	PARTL	$\frac{\partial \tau}{\partial x}$	1	-	the partial derivative of the normalized time variable with respect to X

Subroutines Required: None

Functions Required:
cos (cosine)
sin (sine)
cosh (hyperbolic cosine)
sinh (hyperbolic sine)
SQRT (square root)

Approximate Deck Length 120 (decimal)

```
      PART          -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      FUNCTION PARTL(X)                                        PART0020
C                                                              PART0030
      DIMENSION  WRK(1)                                        PART0040
C                                                              PART0050
      COMMON DATA                                              PART0060
C                                                              PART0070
      EQUIVALENCE          (DATA(391),WRK  )                   PART0080
C                                                              PART0090
      DIMENSION ELEM(4)                                        PART0100
C                                                              PART0110
      EQUIVALENCE          (WRK(130),ELEM  )                   PART0120
      A=ELEM(3)                                                PART0130
      IF(A) 10,10,20                                           PART0140
   20 F1 = SQRT(A)                                             PART0150    4
      F2 = A*SIN(X)                                            PART0160    5
      F3 = SQRT(A*A*A) * (1. -COS(X))                          PART0170    6    7
      GO TO 30                                                 PART0180
   10 F1 =-SQRT(-A)                                            PART0190    10
      F2 =-A*SINH(X)                                           PART0200    11
      F3 =-SQRT(-A*A*A) *(1.-COSH(X))                          PART0210    12   13
   30 PARTL = ELEM(1)*F1 + ELEM(2)* F2 + ELEM(4)* F3           PART0220
      RETURN                                                   PART0230
      END                                                      PART0240
```

STORAGE MAP

FUNCTION PARTL TYPE R

COMMON VARIABLES

COMMON BLOCK // ORIGIN 0C001 LENGTH 01013

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DATA	00000	R	WRK	00606	R	ELEM	01007	R

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
F.000C	01014	R	A	01015	R	F1	01016	R
F2	01017	R	F3	01020	R			

ENTRY POINTS

PARTL SECTION 4

SUBROUTINES CALLED

SQRT	SECTION 5	SIN	SECTION	6	COS	SECTION	7
SINH	SECTION 8	COSH	SECTION	9	SYSLOC	SECTION	10

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
10	9A	01076	20	3A	01036	30	14A	01144

DECK LENGTH IN OCTAL IS 00167.

2.3.1.2 MOTION and the Motion Group

This group consists of those routines designed to evaluate the acceleration vector in the mean equation of 1950.0 frame (selected to eliminate terms in the equations of motion resulting from rotating coordinate systems). Forces resulting from

1) the earth's oblateness
2) atmospheric drag
3) displacement from the reference conic
4) solar radiation pressure
5) solar-lunar gravitational forces

are all included and the group is designed to be complete to itself in that it contains all routines (save general purpose math routines) necessary to evaluate the forces in question (i.e., an atmospheric routine, an ephemeris routine and a solar power function). These routines will be discussed on the following pages.

Subroutine MOTION

Purpose: MOTION serves as the driver routine for all of the previously discussed routines in the Motion Group (i.e., it is designed to compute the differential acceleration vector which will be integrated to define the trajectory as a function of time).

Deck Name: MOTN

Calling Sequence: Call MOTION (ISTART, INDEX)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
O	RCO VCO	$\vec{r}_{c_0}$ $\vec{v}_{c_0}$	3 3	WRK (28) WRK (31)	the conic position and velocity vectors to be utilized in subsequent computations of the reference trajectory if the orbit is rectified in FNCKF (Km, Km/sec)
O	TWCON TFCON	t	1 1	WRK (34) WRK (35)	the time of the last rectification of the conic reference (Julian date in days)
I	R V	$\vec{r}$ $\vec{v}$	3 3	WRK (44) WRK (47)	radius and velocity vectors on the true trajectory (Km and Km/sec) in the frame of 1950.0
I	TW TF	t	1 1	WRK (50) WRK (51)	whole and fractional part of the date (days) relative to J.D. 2433282.423
I	DR DRDOT RDD	$\Delta\vec{r}$ $\Delta\dot{\vec{r}}$ $\Delta\ddot{\vec{r}}$	3 3 3	WRK (52) WRK (55) WRK (58)	displacement, velocity and acceleration vectors relative to the reference conic in the frame of 1950.0 (Km,Km/sec)

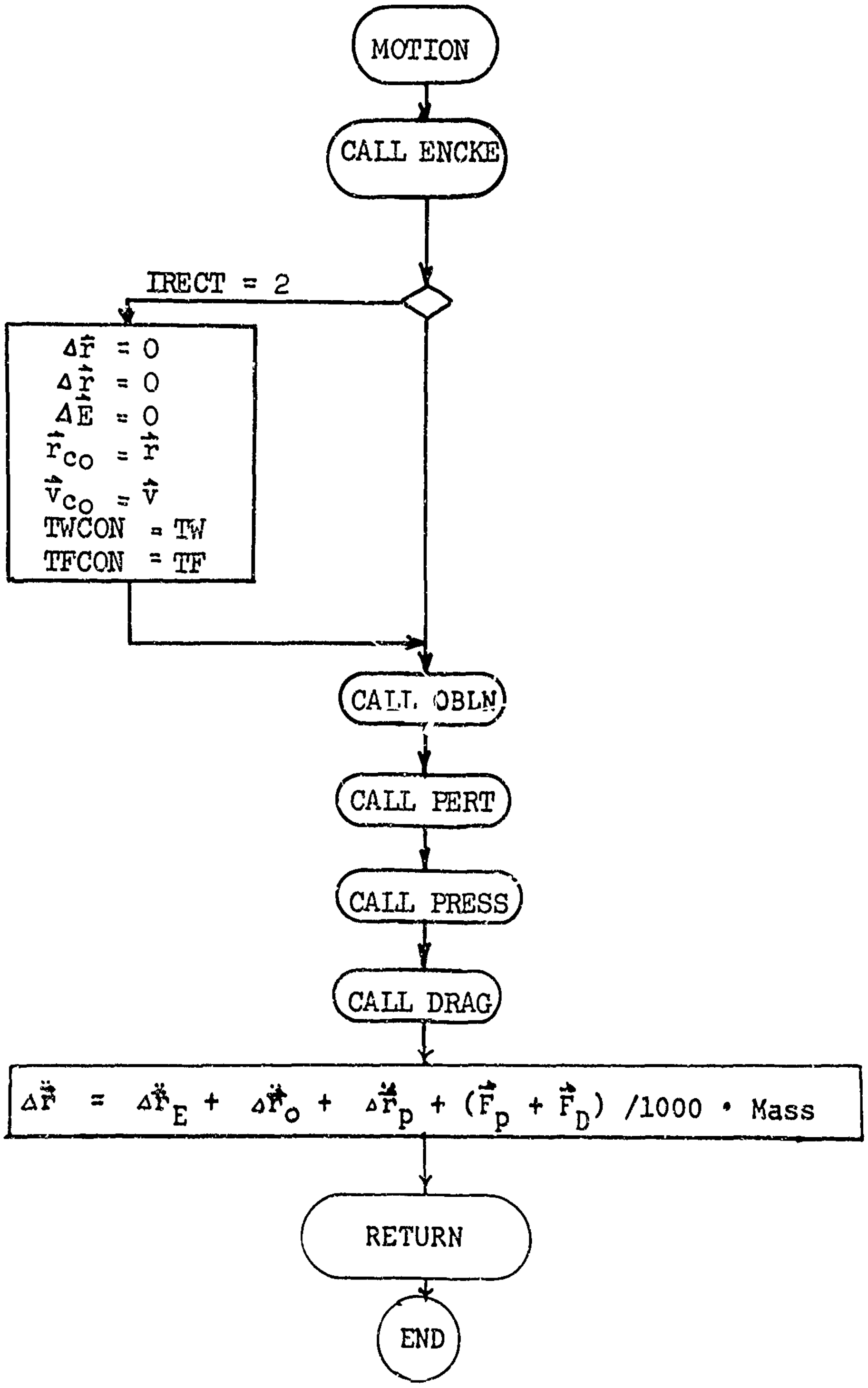
I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	RCONIC	$\vec{r}_c$	3	WRK (117)	radius and velocity vectors defining the present estimate of the conic reference path (Km, Km/sec)
	VCONIC	$\vec{v}_c$	3	WRK (120)	
O	ISTART	-	1	ARG	counter which is equal to 1 set upon rectifying the conic reference to restart the numerical integration
I	INDEX	-	1	ARG	counter used to identify the source of the call of motion (TRAJ or START)

Discussion:

No discussion of MOTION as an independent routine is considered essential except insofar as two points are concerned. First, upon exit from ENCKE, a test is made to determine if ENCKE considered the displacement from the conic reference to be excessive. If so, the conic is rectified, the differential position and velocity vectors are zeroed, the epoch is recorded as the time of rectification, and the ENCKE acceleration is zeroed prior to evaluating all of the other accelerations.

The other point is that since the forces due to drag and solar radiation pressure are expressed in newtons and since the accelerations are being evaluated in Km/sec^2 rather than m/sec^2, the resultant force vector must be divided by 1,000 times the mass to obtain consistent units.

Computational Logic:



```
      MTN          -  EFN    SOURCE STATEMENT  -  IFN(S)  -

      SUBROUTINE MOTION(ISTART,INDEX)                                   MOTN0020
C                                                                       MOTN0030
C     MOTION SERVES AS THE DRIVER ROUTINE IN THE COMPUTATION OF ALL     MOTN0040
C             ACCELERATIONS BEING EXPERIENCED. THE ROUTINE PRESENTLY    MOTN0050
C             INCLUDES THE FORCES RESULTING FROM MOTION OFF THE         MOTN0060
C             REFFERENCE TRAJECTORY(ENCKE), THE SUN AND MOON(PERT),     MOTN0070
C             THE EARTHS OBLATENESS(OBLN), SOLAR RADIATION PRESSURE     MOTN0080
C             (PRESS) AND ATMOSPHERIC DRAG . OTHER FORCES CAN BE ADDED  MOTN0090
C             WHEN THEY CAN BE DESCRIBED ( METEOROID IMPACT,MAGNETIC    MOTN0100
C             DRAG ,HIGHER ORDER TERMS IN THE EARTHS POTENTIAL ).       MOTN0110
C     ALL FORCES ARE COMPUTED IN THE COORDINATE FRAME OF 1950.0 .       MOTN0120
C             FOR THIS REASON, TRANSFORMATION OF THE FRAME IS NOT       MOTN0130
C             NECESSARY UNLESS DATA IS DESIRED IN ANOTHER FRAME (FOR    MOTN0140
C             EXAMPLE ,TOPOCENTRIC OR EQUATOR OF DATE )                 MOTN0150
C     THE ACCELERATION VECTOR HAS THE UNITS OF ** KM/SEC**2 THUS        MOTN0160
C             ALL DISTANCES ARE INPUT IN KM AND VELOCITIES IN KM/SEC.   MOTN0170
C             SATELLITE AREA IS IN METERS**2 AND MASS IS IN KG.         MOTN0180
C     R   = RADIUS VECTOR     (1950.0)                                  MOTN0190
C     V   = VELOCITY VECTOR   (1950.0)                                  MOTN0200
C     DR  = DISPLACEMENT VECTOR RELATIVE TO CONIC REFERENCE.            MOTN0210
C     IRECT= SIGNAL TO CHANGE CONIC REFFERENCE ORBIT                    MOTN0220
C     OMEGA=ANGULAR VELOCITY OF ATMOSPHERE                              MOTN0230
C     SMASS=SATELLITE MASS                                              MOTN0240
C     RDD  =ACCELERATION VECTOR RELATIVE TO NOMINAL TRAJECTORY          MOTN0250
C                                                                       MOTN0260
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * MOTN0270
C                                                                       MOTN0280
      DIMENSION  CON(1), SAT(1), SDA(1), WRK(1), STT(1)                 MOTN0290
C                                                                       MOTN0300
      COMMON DATA                                                       MOTN0310
C                                                                       MOTN0320
      EQUIVALENCE  (DATA(  1),CON), (DATA( 16),SAT), (DATA( 36),SDA)    MOTN0330
     2             ,(DATA(286),STT), (DATA(391),WRK)                    MOTN0340
C                                                                       MOTN0350
      DIMENSION  RCO(3)     ,VCO(3)     ,R(3)       ,V(3)       ,DR(3)  MOTN0360
     2,DRDOT(3) ,RDD(3)     ,RCONIC(3)  ,VCONIC(3)  ,AE(3)      ,AOB(3) MOTN0370
```

```
      MTN          -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      3,AP(3)    ,FS(3)     ,FD(3)                                       MOTN0380
C                                                                        MOTN0390
      EQUIVALENCE          (SAT(  1),SMASS )                             MOTN0400
C                                                                        MOTN0410
      EQUIVALENCE                                                        MOTN0420
     2  (WRK( 28),RCO    ), (WRK( 31),VCO    ), (WRK( 34),TWCON )       MOTN0430
     3 ,(WRK( 35),TFCON ), (WRK( 44),R      ), (WRK( 47),V     )        MOTN0440
     4 ,(WRK( 52),DR     ), (WRK( 55),DRDOT ), (WRK( 58),RDD   )        MOTN0450
     5 ,(WRK(117),RCONIC), (WRK(120),VCONIC), (WRK( 50),TW    )         MOTN0460
     6                                       , (WRK( 51),TF    )        MOTN0470
C                                                                        MOTN0480
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    MOTN0490
C                                                                        MOTN0500
      DATA IRECT/1/                                                      MOTN0510
C                                                                        MOTN0520
C                                                                        MOTN0530
C     COMPUTE ENCKE ACCELERATION AND TEST FOR RECTIFICATION * * * * * *  MOTN0540
      CALL ENCKE(AE,IRECT)                                               MOTN0550    2
      IF(INDEX-1) 10,10,20                                               MOTN0555
   10 IF(IRECT-1) 20,20,30                                               MOTN0560
   30 DO 31 I=1,3                                                        MOTN0570
      DR(I) = 0.                                                         MOTN0580
      DRDOT(I) = 0.                                                      MOTN0590
      AE(I) = 0.                                                         MOTN0600
      RCO(I) = R(I)                                                      MOTN0610
      VCO(I) = V(I)                                                      MOTN0620
      RCONIC(I)=R(I)                                                     MOTN0630
   31 VCONIC(I)=V(I)                                                     MOTN0640
      TWCON = TW                                                         MOTN0650
      TFCON = TF                                                         MOTN0660
      IRECT = 1                                                          MOTN0670
      ISTART = 1                                                         MOTN0675
C                                                                        MOTN0680
C                                                                        MOTN0690
C     COMPUTE BALANCE OF ACCELERATION COMPONENTS * * * * * * * * * * * * *MOTN0700
   20 CALL OBLN(AOB)                                                     MOTN0710   27
      CALL PERT(AP)                                                      MOTN0720   29
```

```
 ****                                                          01/20/86
      MTN        -  EFN   SOURCE STATEMENT  -  IFN(S)  -

   CALL PRESS(FS)                                              MOTN0730    31
   CALL DRAG(FD)                                               MOTN0740    33
   DO 1 I=1,3                                                  MOTN0750
 1 PDD(I) = AE(I) + AOB(I)+ AP(I) +( FS(I)+FD(I) )/(SMASS*1000.)  MOTN0760
   RETURN                                                      MOTN0770
   END                                                         MOTN0780
```

SUBROUTINE MOTION

COMMON VARIABLES

COMMON BLOCK // ORIGIN 00001 LENGTH 01000

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DATA	00000	R	DATA	00000	R	CON	00000	R
SAT	00017	R	SDA	00043	R	STT	00435	R
WRK	00606	R	SMASS	00017	R	RCO	00641	R
VCO	00644	R	TWCON	00647	R	TFCON	00650	R
R	00661	R	V	00664	R	DR	00671	R
DRDOT	00674	R	RDD	00677	R	RCONIC	00772	R
VCONIC	00775	R	TW	00667	R	TF	00670	R

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
AE	01001	R	AOB	01004	R	AP	01007	R
FS	01012	R	FD	01015	R			

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
IRECT	01020	I						

ENTRY POINTS

MOTION SECTION 4

SUBROUTINES CALLED

ENCKE	SECTION	5	OBLN	SECTION	6	PERT	SECTION	7
PRESS	SECTION	8	DRAG	SECTION	9	SYSLOC	SECTION	10

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
10	5A	01041	20	26A	01073	30	7A	01045
31	20A	01057	1	38A	01114			

DECK LENGTH IN OCTAL IS 00156.

2.3.1.2.1 Subroutine OBLN*

Purpose: To compute the non-central nature of the force (expressed in the mean equator of 1950.0 frame) exerted on the satellite resulting from the second, third and fourth spherical harmonics of the Earth's potential function.

Deck Name: OBLN

Calling Sequence: Call OBLN (ACCEL)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
O	ACCEL	$\vec{F}'$	3	Arg	the difference in the force per unit mass exerted by the model of the Earth and the central force for the same satellite position
I	RE	R_e	1	CON (1)	equatorial radius of the Earth (Km)
I	CJ,CH,CD	J,H,D	1,1,1	CON (3,4,5)	second, third and fourth coefficient of Jeffrey's potential function
I	GM	μ	1	CON (6)	gravitational constant for the Earth (Km^3/sec^2)
I	AN	NP	3 x 3	WRK (1)	the rotational matrix relating the frame of 1950.0 to the true equator of date
I	RVEC	$\vec{r}$	3	WRK (44)	radius vector in the frame of 1950.0

* Note: This routine is a version of a routine by the same name originally prepared by JPL (one reference is JPL TR 32-223)

Subroutines Required: None

Functions Required: SQRT

Approximate Deck Length: 250 (decimal)

Formulation:

The potential function (Jeffrey's notation) for a vehicle of unit mass moving in the vicinity of an oblate Earth is

$$U = -\frac{\mu}{r}\left\{\frac{JR_e^2}{3r^2}\left(1-3\sin^2 L\right)+\frac{HR_e^3}{5r^3}\left(3-5\sin^2 L\right)\sin L + \frac{DR_e^4}{35r^4}\left(3-30\sin^2 L+35\sin^4 L\right)+\ldots\right\}$$

where $\vec{r} = r\,\hat{r}$ = position vector in the true equator of date frame

L = geodetic latitude = $\sin^{-1}(\hat{r}_3) \equiv \sin^{-1}\left(\frac{z}{r}\right)$

and the force per unit mass may be evaluated by constructing the negative gradient of U

$$F = -\nabla U = -\left(\frac{\partial U}{\partial x}, \frac{\partial U}{\partial y}, \frac{\partial U}{\partial z}\right)$$

However, the force vector $(\vec{F})$ would be expressed in the true equator of date frame rather than in the desired frame of 1950.0. Thus, before constructing the partials, it is noted that the radius rector in the two frames are related as follows:

$$\vec{r} = A_r\vec{R} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \vec{R}$$

$$R = r$$

and thus that

$$\sin L = \frac{1}{R}\left(a_{31} R_1 + a_{32} R_2 + a_{33} R_3\right)$$

With these substitutions established, the force vector $(\vec{F}\,')$ expressed in the frame of 1950.0 is

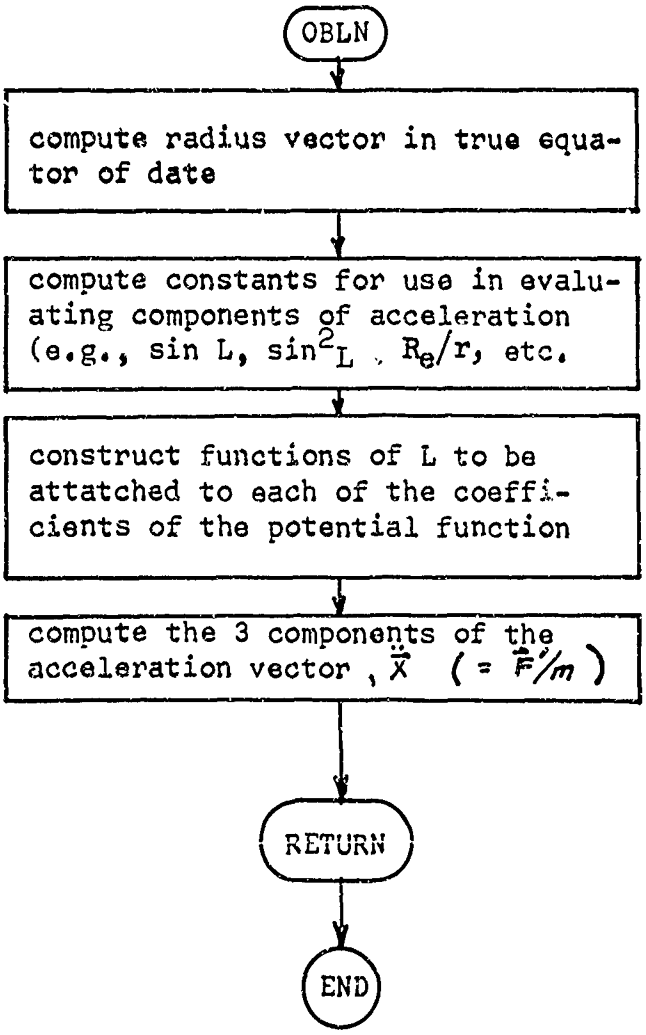
$$\vec{F}\,' = -\left(\frac{\partial U}{\partial R_1}, \frac{\partial U}{\partial R_2}, \frac{\partial U}{\partial R_3}\right)$$

Where

$$-\frac{\partial U}{\partial u_i} = -\frac{J\mu}{R^4} R_e^2 \left\{ \left(1 - \frac{5Z^2}{R^2}\right) \frac{u_i}{R} + 2\frac{Z}{R} a_{3i} \right\}$$
$$- \frac{H\mu}{R^3} R_e^3 \left\{ \left(3 - 7\frac{Z^2}{R^2}\right) \frac{Z}{R^2} u_i + \left(-\frac{3}{5} + 3\frac{Z^2}{R^2}\right) a_{3i} \right\}$$
$$- \frac{D\mu}{R^6} R_e^4 \left\{ \left(\frac{3}{7} - 6\frac{Z^2}{R^2} + 9\frac{Z^4}{R^4}\right) \frac{u_i}{R} + \left(\frac{12}{7} - \frac{4Z^2}{R^2}\right) \frac{Z a_{3i}}{R} \right\}$$

$u_i = R_1$ or R_2 or R_3

Computational Logic:



```
      SUBRØUTINE ØBLN(ACCEL)                                            ØBLN0020
C                                                                       ØBLN0030
C     THIS RØUTINE CØMPUTES THE ØBLATENESS ACCELERATIØN ACTING ØN A     ØBLN0040
C             SATELLITE MØVING IN THE FØRCE FIELD ØF A BØDY DESCRIBED   ØBLN0050
C             BY JEFFRIES PØTENTIAL FUNCTIØN . IT CAN ALSØ CØMPUTE THE  ØBLN0060
C     RVEC    = INSTANTANEØUS RADIUS VECTØR IN FRAME ØF 1950.0          ØBLN0070
C     GM      = GRAVITATIØNAL CØNSTANT                                  ØBLN0080
C     RE      = EQUATØRIAL RADIUS ØF CENTRAL BØDY                       ØBLN0090
C     CJ,CH,CD=CØEFFICIENTS ØF THE SECØND,THIRD AND FØURTH HARMØNICS    ØBLN0100
C     ACCEL   = PERTURBING ACCELERATIØN VECTØR IN FRAME ØF 1950.0       ØBLN0110
C     AN      = INPUT ARRAY TØ CØNVERT PERTURBATIØN TØ FRAME ØF DATE    ØBLN0120
C                                                                       ØBLN0130
C                                                                       ØBLN0140
      DIMENSIØN  CØN(1), SAT(1), SDA(1), WRK(1), STT(1)                 ØBLN0150
      CØMMØN DATA                                                       ØBLN0160
      EQUIVALENCE  (DATA(  1),CØN), (DATA( 15),SAT), (DATA( 36),SDA)    ØBLN0170
     2             ,(DATA(286),STT), (DATA(391),WRK)                    ØBLN0180
C                                                                       ØBLN0190
      DIMENSIØN   AN(3,3)  ,RVEC(3)  ,ACCEL(3)  ,X(3)                   ØBLN0200
C                                                                       ØBLN0210
      EQUIVALENCE           (CØN(  1),RE    ), (CØN(  3),CJ    )        ØBLN0220
     2  ,(CØN(  4),CH    ), (CØN(  5),CD    ), (CØN(  6),GM    )        ØBLN0230
C                                                                       ØBLN0240
      EQUIVALENCE           (WRK(  1),AN    ), (WRK( 44),RVEC  )        ØBLN0250
C                                                                       ØBLN0260
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   ØBLN0270
C                                                                       ØBLN0280
      DØ 2 I=1,3                                                        ØBLN0290
    5 X(I)= 0.                                                          ØBLN0300
      DØ 2 J=1,3                                                        ØBLN0310
    2 X(I) = X(I) + AN(I,J)*RVEC(J)                                     ØBLN0320
   10 R2 = X(1)**2+X(2)**2+X(3)**2                                      ØBLN0330
   15 R = SQRT (R2)                                                     ØBLN0340   19
   20 UR2 =GM/R2                                                        ØBLN0350
   25 AR =RE/R                                                          ØBLN0360
   30 AR2 = AR*AR                                                       ØBLN0370
```

```
      ØBL          -  EFN   SØURCE STATEMENT  -  IFN(S)  -

 35 AR3 = AR*AR2                                              ØBLV0380
 40 AR4 = AR*AR3                                              ØBLV0390
 45 ZR = X(3)/R                                               ØBLV0400
 50 ZR2 = ZR*ZR                                               ØBLN0410
 55 ZR4 = ZR2*ZR2                                             ØBLV0420
 60 TM1 = (1.-5.*ZR2)/R                                       ØBLV0430
 65 TM2 = 2.*ZR                                               ØBLV0440
 70 TM3 = (3.-7.*ZR2)*ZR/R                                    ØBLV0450
 75 TM4 = -.6+3.*ZR2                                          ØBLV0460
 80 TM5 = (.42857143-6.*ZR2+9.*ZR4)/R                         ØBLN0470
 85 TM6 = (1.7142857-4.*ZR2)*ZR                               ØBLN0480
 90 TØ1 = -CJ*UR2*AR2                                         ØBLV0490
 95 TØ2 =-CH*UR2*AR3                                          ØBLV0500
 98 TØ3 =-CD*UR2*AR4                                          ØBLV0510
    DØ 1 I=1,3                                                ØBLV0520
  1 ACCEL(I)=TØ1*(TM1*RVEC(I)+TM2*AN(3,I))                    ØBLV0530
   1        +TØ2*(TM3*RVEC(I)+TM4*AN(3,I))                    ØBLN0540
   2        +TØ3*(TM5*RVEC(I)+TM6*AN(3,I))                    ØBLV0550
    RETURN                                                    ØBLV0560
    END                                                       ØBLV0570
```

STORAGE MAP

SUBROUTINE OBLN

COMMON VARIABLES

COMMON BLOCK // ORIGIN 00001 LENGTH 00664

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DATA	00000	R	DATA	00000	R	CON	00000	R
SAT	00017	R	SDA	00043	R	STT	00435	R
WRK	00606	R	RE	00000	R	CJ	00002	R
CH	00003	R	CD	00004	R	GM	00005	R
AN	00606	R	RVEC	00651	R			

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
X	00665	R						

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
I	00670	I	R2	00671	R	R	00672	R
UR2	00673	R	AR	00674	R	AR2	00675	R
AR3	00676	R	AR4	00677	R	ZR	00700	R
ZR2	00701	R	ZR4	00702	R	TM1	00703	R
TM2	00704	R	TM3	00705	R	TM4	00706	R
TM5	00707	R	TM6	00710	R	TO1	00711	R
TO2	00712	R	TO3	00713	R			

ENTRY POINTS

OBLN SECTION 4

SUBRØUTINES CALLED

SQRT SECTIØN 5 SYSLØC SECTIØN 6

EFN IFN CØRRESPØNDENCE

EFN	IFN	LØCATIØN	EFN	IFN	LØCATIØN	EFN	IFN	LØCATIØN
2	9A	00754	5	5A	00742	10	17A	00765
15	18A	01000	20	20A	01005	25	21A	01010
30	22A	01013	35	23A	01016	40	24A	01021
45	25A	01024	50	26A	01027	55	27A	01032
60	28A	01035	65	29A	01043	70	30A	01046
75	31A	01056	80	32A	01062	85	33A	01074
90	34A	01103	95	35A	01111	98	36A	01117
1	40A	01127						

DECK LENGTH IN ØCTAL IS 00335.

2.3.1.2.2 Subroutine DRAG

Purpose: To evaluate the force acting on a satellite (in the mean equator of 1950.0 frame) resulting from passage through a rotating oblate atmosphere.

Deck Name: DRAG

Calling Sequence: Call DRAG (DF)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	OMEGA	ω	1	CON (7)	Spin rate of the Earth
I	A	A	1	SAT (2)	Reference area
I	CD	C_D	1	SAT (3)	Drag coefficient
I	ROTATE	T^{-1}	3 x 3	WRK (1)	Rotational matrices relating vectors in
	ROTINV	T	3 x 3	WRK (10)	the true equator of date frame to 1950.0
I	R	$\vec{r}$	3	WRK (44)	Radius vector (1950.0)
I	V	$\vec{v}$	3	WRK (47)	Velocity vector (1950.0)
I	TW	t	1	WRK (50)	Day number relative to 1950.0 (used as input into atmospheric density routine)
O	DF	$\vec{D}$	3	Arg	Drag force (if A = square meters, density = Kg/m^3, and velocity = Km/sec, then D = newtons)

Subroutines Required: ATMS (atmospheric density), CROSS (cross-product)

Functions Required: AMAG (vector magnitude)

Approximate Deck Length: 146 (octal)

Formulation:

The drag force is computed in the frame of 1950.0 as follows

$$\vec{\omega}' = [T]\,\vec{\omega}$$

where [T] = rotational transformation relating the true equator of date to the mean equator of 1950.0

$\vec{\omega}$ = spin vector of the Earth in the true equator of date frame

$\vec{\omega}'$ = spin vector in the frame of 1950.0

$$\vec{V}_\omega = \vec{\omega}' \times \vec{r}$$

$$\vec{V}_r = \vec{V} - \vec{V}_\omega$$

where $\vec{r}$ = radius vector (1950.0)

$\vec{V}$ = velocity vector (1950.0)

$\vec{V}_r$ = velocity relative to the wind (1950.0)

$\vec{V}_\omega$ = velocity of the wind (1950.0)

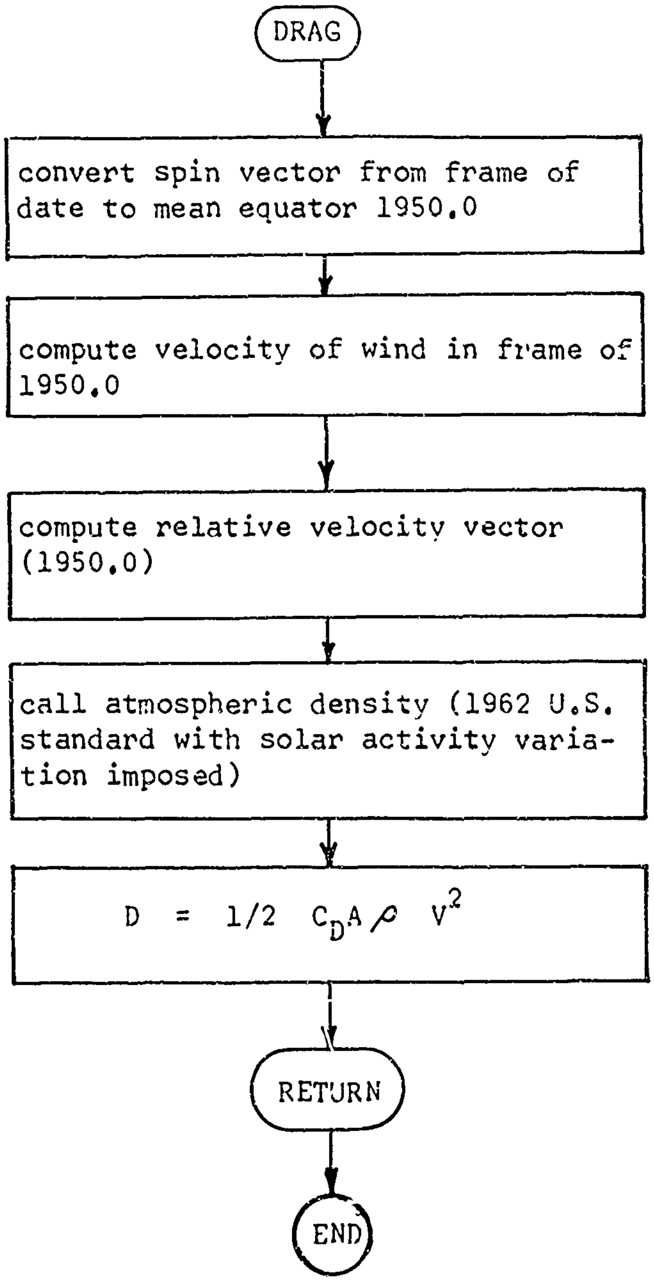
$$\vec{D} = -\tfrac{1}{2}\rho(\vec{r}, T)\, C_D A \;\vec{V}_r\, |\vec{V}_r|$$

where $\vec{D}$ = drag force in newtons

$\rho(\vec{r}, T)$ = mass density of the atmosphere

$C_D A$ = drag coefficient times reference area

Computational Logic:



```
      SUBROUTINE DRAG(DF)                                              DRAG0020
C                                                                      DRAG0030
C     THIS ROUTINE COMPUTES DRAG FORCES  EXERTED ON A BODY MOVING IN   DRAG0040
C             AN ATMOSPHERE WHICH CAN BE DEFINED AS A FUNCTION OF      DRAG0050
C             TIME(MEASURED FROM 1950) OR POSITION(IN THE TRUE EQUATOR DRAG0060
C             OF DATE FRAME )                                          DRAG0070
C     R      = POSITION VECTOR IN FRAME OF 1950.0  ( KM )              DRAG0080
C     V      = VELOCITY VECTOR IN FRAME OF 1950.0  ( KM/SEC )          DRAG0090
C     CD     = COEFFICIENT OF DRAG                                     DRAG0100
C     A1     = CROSS SECTIONAL AREA OF SPHERICAL SATELLITE  ( M**2 )   DRAG0110
C     OMEGA  = ANGULAR VELOCITY OF CENTRAL BODY                        DRAG0120
C     ROTINV = INVERSE OF MATRIX RELATING EQUATOR OF 1950. TO DATE     DRAG0130
C     TW     = JULIAN DATE                                             DRAG0140
C     DF     = DRAG FORCE IN THE FRAME OF 1950.   ( NEWTONS )          DRAG0150
C                                                                      DRAG0160
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  DRAG0170
C                                                                      DRAG0180
      DIMENSION  CON(1), SAT(1), SDA(1), WRK(1), STT(1)                DRAG0190
C                                                                      DRAG0200
      COMMON DATA                                                      DRAG0210
C                                                                      DRAG0220
      EQUIVALENCE  (DATA(  1),CON), (DATA( 15),SAT), (DATA( 36),SDA)   DRAG0230
     2             ,(DATA(286),STT), (DATA(391),WRK)                   DRAG0240
C                                                                      DRAG0250
C                                                                      DRAG0260
      DIMENSION  R(3)       ,V(3)       ,RE(3)      ,W(3)       ,VW(3) DRAG0270
     2           ,DF(3)      ,VREL(3)    ,ROTINV(3,3)                  DRAG0280
C                                                                      DRAG0290
      EQUIVALENCE            (CON(  7),OMEGA ), (SAT(  2),A1     )     DRAG0300
     2  ,(SAT(  3),CD     ), (WRK( 50),TW     ), (WRK(  1),ROTATE)     DRAG0310
     3  ,(WRK( 10),ROTINV), (WRK( 44),R      ), (WRK( 47),V      )     DRAG0320
C                                                                      DRAG0330
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  DRAG0340
C                                                                      DRAG0350
C     THE VELOCITY OF THE RELATIVE WIND                                DRAG0360
      W(1) = OMEGA*ROTINV(1,3)                                         DRAG0370
```

```
     DRAGG        -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      W(2) = OMEGA*ROTINV(2,3)                                    DRAG0380
      W(3) = OMEGA*ROTINV(3,3)                                    DRAG0390
      CALL CROSS(W,R ,VW)                                         DRAG0400     2
      DO 1 I=1,3                                                  DRAG0410
    1 VREL(I)= V(I)-VW(I)                                         DRAG0420
      VMAG = AMAG(VREL)                                           DRAG0430
C                                                                 DRAG0440
C     THE ATMOSPHERE WILL NOW BE CALLED AND THE DENSITY COMPUTED  DRAG0450
C              IN KG/M**3 . DRAG VECTOR THEN DEFINED IN NEWTONS . DRAG0460    13
      CALL MATMPY( ROTATE,3,3,R,3,1,RE )                          DRAG0470    14
      CALL ATMS( RE,TW,DENS )                                     DRAG0480    16
      CONST = .5 * DENS * CD * A1                                 DRAG0490
      DO 3 I = 1,3                                                DRAG0500
    3 DF(I)  = -CONST * VREL(I) * VMAG * 1.E5                     DRAG0510
      RETURN                                                      DRAG0520
      END                                                         DRAG0530
```

FS305 DRAGG STORAGE MAP 11/24/85 PAGE 40

SUBROUTINE DRAG

COMMON VARIABLES

COMMON BLOCK // ORIGIN 00001 LENGTH 00670

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DATA	00000	R	DATA	00000	R	CON	00000	R
SAT	00017	R	SDA	00043	R	STT	00435	R
WRK	00606	R	OMEGA	00006	R	A1	00020	R
CD	00021	R	TW	00657	R	ROTATE	00606	R
ROTINV	00617	R	R	00651	R	V	00664	R

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
RE	00671	R	W	00674	R	VW	00677	R
VREL	00702	R						

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
VMAG	00705	R	DENS	00706	R	CONST	00707	R

ENTRY POINTS

DRAG SECTION 4

SUBROUTINES CALLED

CROSS	SECTION	5	AMAG	SECTION	6	MATMPY	SECTION	7
ATMS	SECTION	8	SYSLOC	SECTION	9			

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1	7A	00737	3	21A	01001			

DECK LENGTH IN OCTAL IS 00146.

2.3.1.2.2.1 Subroutine ATMS

Purpose: ATMS computes an approximate atmospheric mass density (Kg/m^3) at any altitude (h) between 100 and 700 Km. (Below 100 Km the density is set equal to that of 100 Km and an error message is recorded; above 700 Km the density is set equal to zero). The U. S. Standard Atmosphere 1962, is used as a reference for a table (stored in memory within the Data Input Group) with 21 entries (10 Km steps $100 < h < 200$, 50 Km steps $200 < h < 700$). A dichotomic interpolation is then used in conjunction with the assumed exponential atmosphere (in the neighborhoods of the tabulated point) to produce an interpolated density (2 to 3 place accuracy throughout the table). These data are then corrected for the 11-year sunspot activity cycle.

Deck Name ATMS

Calling Sequence: CALL ATMS (RVEC, TJD, DENS)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	RVEC	$\vec{r}$	3	Arg	radius vector in the true equator of date frame
I	TJD	t	1	Arg	dummy variable (such as the Julian date) which can be used to compute corrections to ρ
O	DENS	ρ	1	Arg	atmospheric mass density (Kg/m^3)
I	ALT 1 ALT 2 ALT 3	h_1 h_2 h_3	1 1 1	ATCON (1) ATCON (2) ATCON (3)	Altitude limits for density data

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	REQT RPOL	r_e r_p	1 1	ATCON (4) ATCON (5)	equatorial and polar radii for the Earth
I	STEP 1 STEP 2	Δh_1 Δh_2	1 1	ATCON (6) ATCON (7)	altitude increment for density and lapse rate tables
I	RHOF RATE	ρ_i K_i	21 21	TABLE (1) TABLE (22)	density and lapse rate at altitudes between h_1 and h_3

Subroutines Required: None

Functions Required: SQRT
EXP
SIN
ALOG
AMAG

Deck Length: 00524 (octal)

Formulation:

Atmospheric density will be approximated utilizing data taken from the 1962 U. S. Standard Atmosphere and assumed exponential behavior with a single parameter, altitude. Thus, the first task is to compute the radial distance from the center of the Earth to the subsatellite point (the intersection of the radius vector to the satellite with the Earth's surface) which will be defined to be r. This task, in turn, will be accomplished by employing the oblate spheroid representation of the Earth.

Let R be the satellite position vector and c be the projection in the plane of the true equator of date

$$r^2 = r_1^2 + r_2^2 + r_3^2$$

$$c^2 = r_1^2 + r_2^2$$

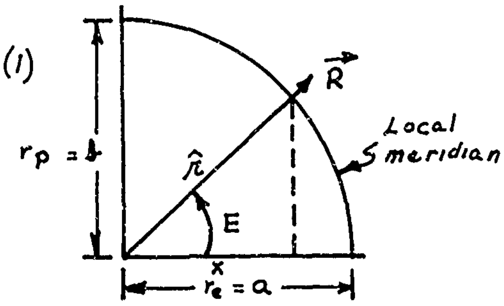
If c = 0, the satellite is directly above either the North or South Pole, and

$$\hat{r} = r_p$$

where, r_p = polar Earth radius and $\hat{r}$ = local Earth radius. However, if $c \neq 0$,

$$\hat{r} = \sqrt{x^2 + y^2} \qquad (1)$$

$$= x\sqrt{1 + \tan^2 E}$$



where $\quad \tan E = \dfrac{r_3}{C}$

Then from the equation for an ellipse,

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

$$\frac{b^2}{a^2} + \tan^2 E = \frac{b^2}{x^2}$$

or

$$x = \frac{b}{\sqrt{\frac{b^2}{a^2} + \tan^2 E}}$$

Substituting into (1)

$$\hat{r} = \frac{b\sqrt{1 + \tan^2 E}}{\sqrt{\frac{b^2}{a^2} + \tan^2 E}}$$

and the height above the Earth is

$$h = r - \hat{r}$$

Now the atmosphere is assumed to be constructed in uniform concentric shells around the (oblate spheroid) Earth with densities which agree with those predicted by the 1962 U. S. Standard Model. For simplicity, a two part table of these data has been prepared. Part one covers altitudes from 100 Km to 200 Km in 10 Km steps while part 2 covers from 200 Km to 700 Km altitude in 50 Km steps (there are, therefore, 21 entries in the table). Since this region contains all altitudes for which the tenuous atmosphere should be considered in studies of satellite motion, the table was not extended. Rather, below 100 Km the density is set equal to the value at 100 Km, and above 700 Km the density is set equal to zero.

For any altitude, h, between 100 Km and 700 Km the values in the table corresponding to the nearest altitude above and below h, must be determined and interpolated to find the density at h. Let m be the index corresponding to the nearest tabulated altitude below h, then,

(1) between 100 and 200 Km, m = greatest integer $\left(\frac{h-100}{10}\right) + 1$

(2) between 200 and 700 Km, m = greatest integer $\left(\frac{h-200}{50}\right) + 11$

Let Δh be the distance between h and the nearest tabulated altitude below h,

(1) if $100 \text{ Km} \leq h \leq 200 \text{ Km}$

$$\left.\begin{aligned} X &= \text{greatest integer in } \frac{h-100}{10} \\ \Delta h &= (h-100) - X\,(10) \end{aligned}\right\}$$

(2) if $200 \text{ Km} \leq h \leq 700 \text{ Km}$

$$\left.\begin{aligned} X &= \text{greatest integer in } \left(\frac{h-200}{50}\right) \\ \Delta h &= (h-200) - X\,(50) \end{aligned}\right\}$$

The tabular values of density above and below h can now be extrapolated to predict two values of the density at h; then, an interpolation can be made between the two predicted values of density to include the non-exponential nature of the atmosphere.

Let $\Delta\tilde{h}$ be the distance between h and the nearest tabulated altitude above h, then

$$\Delta\tilde{h} = \Delta h - 10 \qquad \text{if } 100 \leq h \leq 200$$

$$\Delta\tilde{h} = \Delta h - 50 \qquad \text{if } 200 \leq h \leq 700$$

and let ρ_1 = extrapolation of lower value (index m) and ρ_2 = extrapolation of upper value (index m + 1)

$$\rho_1 = \tilde{\rho}_m \, e^{-r_m \Delta h}$$

$$\rho_2 = \tilde{\rho}_{m+1} \, e^{-(r_{m+1}) \Delta \tilde{h}}$$

where $\tilde{\rho}$ = tabulated value of density

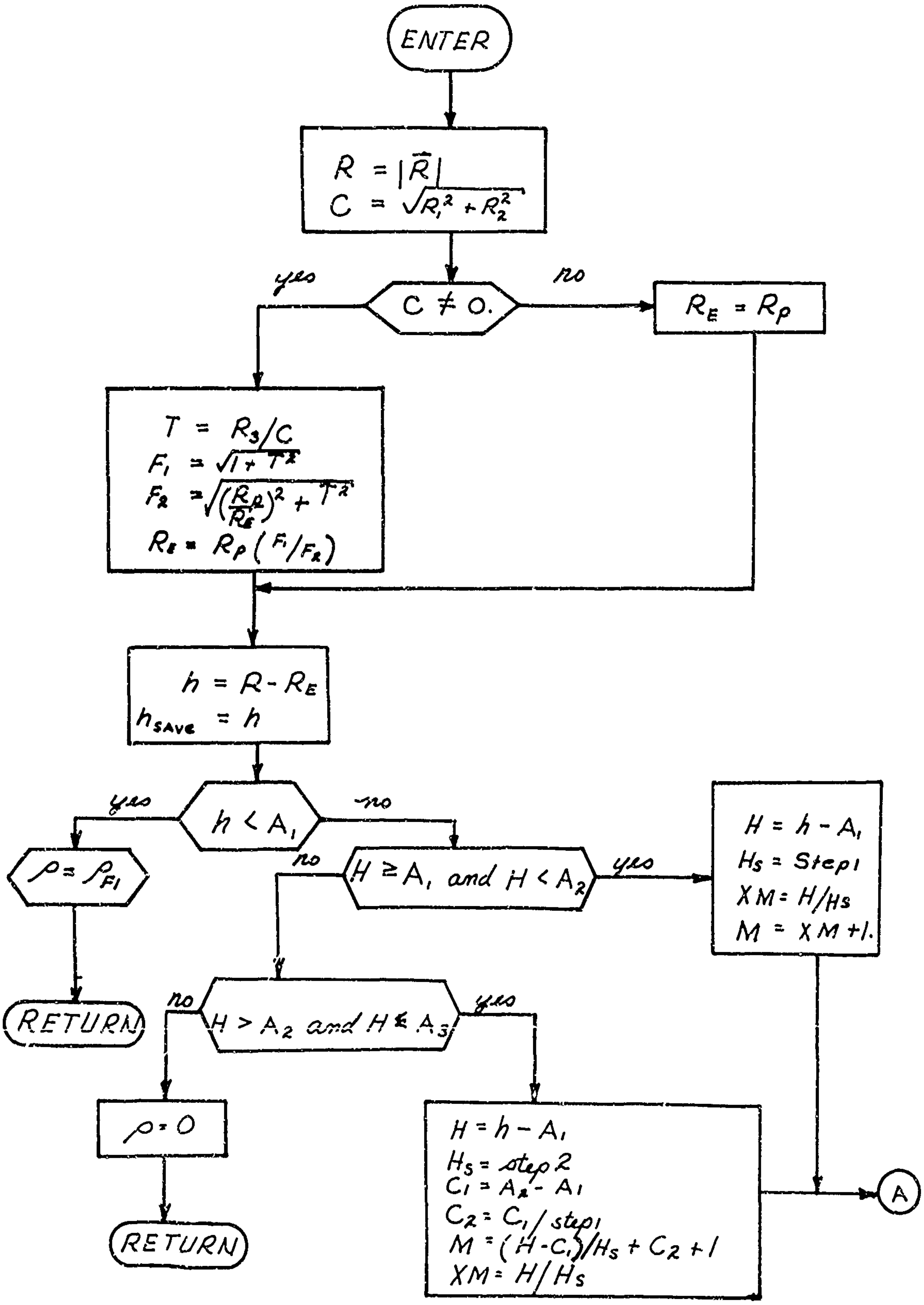
r = tabulated slope (lapse rate) $= \frac{d\rho}{dh}$

Finally, by linear interpolation between

$$\rho = y\rho_2 + (1-y)\rho_1$$

$$y = \Delta h/10 \text{ or } \Delta h/50$$

Computational Logic:



A

$KC = XM$

$XM = KC$

$\Delta h = h - XM * H_s$

$\Delta 2 = \Delta h - H_s$

$N = M + 1$

$\rho_1 = \rho_F(M) \exp(-\dot{\rho}(M)\Delta h)$

$\rho_2 = \rho_F(N) \exp(-\dot{\rho}(N)\Delta h)$

$xh = \Delta h / H_s$

$\hat{\rho} = xh\,\rho_2 + (1 - xh)\rho_1$

RETURN

END

```
      SUBROUTINE ATMS (RVEC, TJD, DENS)                                 ATMS0020
C        THIS ROUTINE COMPUTES THE ATMOSPHERIC DENSITY IN KILOGRAMS PER ATMS0030
C        METER CUBED, BY FITTING AN EXPONENTIAL TO POINTS TAKEN FROM    ATMS0040
C        THE U. S. STANDARD ATMOSPHERE, 1962.  ABOVE 700 KM THE DENSITY ATMS0050
C        IS SET EQUAL TO ZERO, BELOW 100 KM THE VALUE AT 100 KM IS USED ATMS0060
C        AND AN ERROR MESSAGE IS PRINTED.                               ATMS0070
C                                                                       ATMS0080
      DIMENSION RVEC(3)                                                 ATMS0090
C                                                                       ATMS0100
      COMMON /ATCON/ ALT1, ALT2, ALT3, REQT, RPOL, STEP1, STEP2         ATMS0110
      COMMON /TABLE/ RHOF(21), RATE(21)                                 ATMS0120
C                                                                       ATMS0130
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   ATMS0140
C                                                                       ATMS0150
      RMAG  = AMAG (RVEC)                                               ATMS0160     2
      CONST = SQRT (RVEC(1)**2 + RVEC(2)**2)                            ATMS0170
C                                                                       ATMS0180     3
      IF (CONST .NE. 0.) GO TO 20                                       ATMS0190
      RELLP = RPOL                                                      ATMS0200
      GO TO 25                                                          ATMS0210
C                                                                       ATMS0220
   20 TANE  = RVEC(3) / CONST                                           ATMS0230
      FOX1  = SQRT (1. + TANE**2)                                       ATMS0240     9
      FOX2  = SQRT ((RPOL/REQT)**2 + TANE**2)                           ATMS0250     10
      RELLP = RPOL*FOX1/FOX2                                            ATMS0260
   25 H     = RMAG - RELLP                                              ATMS0270
      HSAVE = H                                                         ATMS0280
      IF  (H .LT. ALT1)  GO TO 30                                       ATMS0290
      IF ((H .GE. ALT1) .AND. (H .LE. ALT2)) GO TO 60                   ATMS0300
      IF ((H .GT. ALT2) .AND. (H .LE. ALT3)) GO TO 70                   ATMS0310
      GO TO 90                                                          ATMS0320
   30 DENS = RHOF(1)                                                    ATMS0330
      WRITE (6, 40)                                                     ATMS0340     23
   40 FORMAT ( /  29H ALTITUDE IS LESS THAN 100 KM   )                  ATMS0350
      GO TO 2000                                                        ATMS0360
C                                                                       ATMS0370
```

```
   60 H      = H - ALT1                                                ATMS0380
      HSTEP = STEP1                                                    ATMS0390
      XM     = H / HSTEP                                               ATMS0400
      M      = XM + 1.                                                 ATMS0410
      GO TO 100                                                        ATMS0420
C                                                                      ATMS0430
   70 H      = H - ALT1                                                ATMS0440
      HSTEP = STEP2                                                    ATMS0450
      CON1   = ALT2 - ALT1                                             ATMS0460
      CON2   = CON1 / STEP1                                            ATMS0470
      M      = (H - CON1) / STEP2 + CON2 + 1.                          ATMS0480
      XM     = H / HSTEP                                               ATMS0490
      GO TO 100                                                        ATMS0500
C                                                                      ATMS0510
   90 DENS = 0.                                                        ATMS0520
      GO TO 2000                                                       ATMS0530
C                                                                      ATMS0540
  100 KC      = XM                                                     ATMS0550
      XM      = KC                                                     ATMS0560
      DELTAH = H - XM*HSTEP                                            ATMS0570
      DELTA2 = DELTAH - HSTEP                                          ATMS0580
      N       = M + 1                                                  ATMS0590
      RHO1    = RHOF(M) * EXP(-RATE(M)*DELTAH)                         ATMS0600     36
      RHO2    = RHOF(N) * EXP(-RATE(N)*DELTA2)                         ATMS0610   . 39
      XH      = DELTAH / HSTEP                                         ATMS0620
      DENS    = XH*RHO2 + (1. - XH)*RHO1                               ATMS0630
C                                                                      ATMS0750
C                                                                      ATMS0760
2000   CONTINUE                                                        ATMS0770
       RETURN                                                          ATMS0770
       END                                                             ATMS0780
```

SUBROUTINE ATMS

COMMON VARIABLES

COMMON BLOCK ATCON   ORIGIN 00001   LENGTH 00007

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
ALT1	00000	R	ALT2	00001	R	ALT3	00002	R
REQT	00003	R	RPOL	00004	R	STEP1	00005	R
STEP2	00006	R						

COMMON BLOCK TABLE   ORIGIN 00010   LENGTH 00052

RHOF	00000	R	RATE	00025	R

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
RMAG	00062	R	CONST	00063	R	RELLP	00064	R
TANE	00065	R	FOX1	00066	R	FOX2	00067	R
H	00070	R	HSAVE	00071	R	HSTEP	00072	R
XM	00073	R	M	00074	I	CON1	00075	R
CON2	00076	R	KC	00077	I	DELTAH	00100	R
DELTA2	00101	R	N	00102	I	RHO1	00103	R
RHO2	00104	R	XH	00105	R			

ENTRY POINTS

ATMS	SECTION	7

SUBROUTINES CALLED

AMAG	SECTION	8	SQRT	SECTION	9	.FWRD.	SECTION	10
EXP	SECTION	11	.UN06.	SECTION	12	.FFIL.	SECTION	13

****

ATMOS STORAGE MAP 12/13/85 PAGE 21

.FCNV.	SECTION	14	E.1	SECTION	15	E.2	SECTION	16
E.3	SECTION	17	E.4	SECTION	18	CC.1	SECTION	19
CC.2	SECTION	20	CC.3	SECTION	21	CC.4	SECTION	22
SYSLOC	SECTION	23						

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
20	8A	00160	25	11A	00217	30	22A	00254
60	25A	00265	70	27A	00304	90	30A	00335
40	FORMAT	00121	2000	41A	00435	100	32A	00337

DECK LENGTH IN OCTAL IS 00413.

2.3.1.2.3 Subroutine ENCKE

Purpose: ENCKE provides the correction to the acceleration vector resulting from motion on the true (rather than the reference) trajectory

Deck Name: ENCK

Calling Sequence: CALL ENCKE (ACCEL, IRECT)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	GM	μ	1	CON (6)	Gravitational constant for the central body
I	RCONIC	$\vec{\rho}$	3	WRK (117)	Position vector on the conic reference trajectory
I	DELTA	$\Delta\vec{r}$	3	WRK (52)	Displacement relative to conic position ($\Delta\vec{r} = \vec{r} - \vec{\rho}$)
O	ACCEL	$\Delta\ddot{\vec{r}}$	3	Arg	Differential acceleration vector
O	IRECT	-	1	Arg	Signal for MOTION to alter the reference trajectory

Subroutine Required: None

Functions Required: DOT (dot product)
SQRT (square root)

Approximate Deck Length: 213 (octal)

Formulation:

Encke's method of integration involves the computation of the acceleration vector relative to a known reference which in the present case is an ellipse. Thus in addition to real perturbing forces there are terms which must be included for the off-nominal nature of the motion. Consider

$$\ddot{\vec{r}} = \frac{-\mu \vec{r}}{r^3} + \vec{F} \tag{1}$$

$$\ddot{\vec{\rho}} = \frac{-\mu \vec{\rho}}{\rho^3} \tag{2}$$

where $\vec{r}$ = radius vector (in some frame) for true trajectory

$\vec{F}$ = summation of all non-central forces

$\vec{\rho}$ = radius vector (at the time corresponding to the instantaneous position on the true trajectory) on the conic reference

defining

$$\Delta\vec{r} = \vec{r} - \vec{\rho} \tag{3}$$

The differential equations for $\Delta\vec{r}$ can be obtained

$$\Delta\ddot{\vec{r}} = F + \mu\left[\frac{\vec{\rho}}{\rho^3} - \frac{\vec{r}}{r^3}\right]$$

But since $|\Delta\vec{r}|$ may be small, the term in the brackets may be known to less accuracy than $\vec{r}$ or $\vec{\rho}$ due to the fact that two nearly equal numbers are differenced. Thus, a modification to maintain accuracy for such cases will be presented.

$$\Delta\ddot{\vec{r}} = \vec{F} + \frac{\mu}{\rho^3}\left[\vec{\rho} - \frac{\rho^3}{r^3}\vec{r}\right]$$

$$\equiv \vec{F} + \frac{\mu}{\rho^3}\left[\vec{\rho} - (1 - f_g)\vec{r}\right] \qquad f_g << 1 \tag{4}$$

$$= \vec{F} + \frac{\mu}{\rho^3}\left[-\Delta\vec{r} + f_g\vec{r}\right]$$

Now f_q has one more degree of freedom than does the ratio $(\rho/r)^3$ so the following identity can be written arbitrarily

$$(1-f_q) = (1+2q)^{-3/2} = (\rho/r)^3 \tag{5}$$

or

$$f_q = 1-(1+2q)^{-3/2} \tag{6a}$$

$$= 3q - \frac{3\cdot 5}{2} q^2 + \frac{3\cdot 5\cdot 7}{2\cdot 3} q^3 - \frac{3\cdot 5\cdot 7\cdot 9}{2\cdot 3\cdot 4} q^4$$

$$+ \tag{6b}$$

where

$$\frac{r^2}{\rho^2} = 1+2q$$

$$q = \frac{r^2-\rho^2}{2\rho^2} \tag{7}$$

$$= \frac{1}{2\rho^2}\left\{(\vec{\rho}+\vec{\Delta r})\cdot(\vec{\rho}+\vec{\Delta r}) - \vec{\rho}\cdot\vec{\rho}\right\}$$

$$= \frac{\vec{\rho}\cdot\vec{\Delta r} + 2\vec{\Delta r}\cdot\vec{\Delta r}}{\vec{\rho}\cdot\vec{\rho}}$$

Equation 6b,when truncated at any given term,defines the accuracy to be obtained from equation 4 for the case where $|\vec{r}| \approx |\vec{\rho}|$. Thus, if a fixed number of terms are consistently carried, the maximum value that q can attain before the reference trajectory is altered (rectified) (or before the equation for evaluating f_q is changed from equation 6b to 6a) is defined by the maximum error allowable in the scalar f_q. For example, if this allowable error is arbitrarily set at 10^{-8}, and if seven terms in f_q are carried, the upper limit in q is defined by

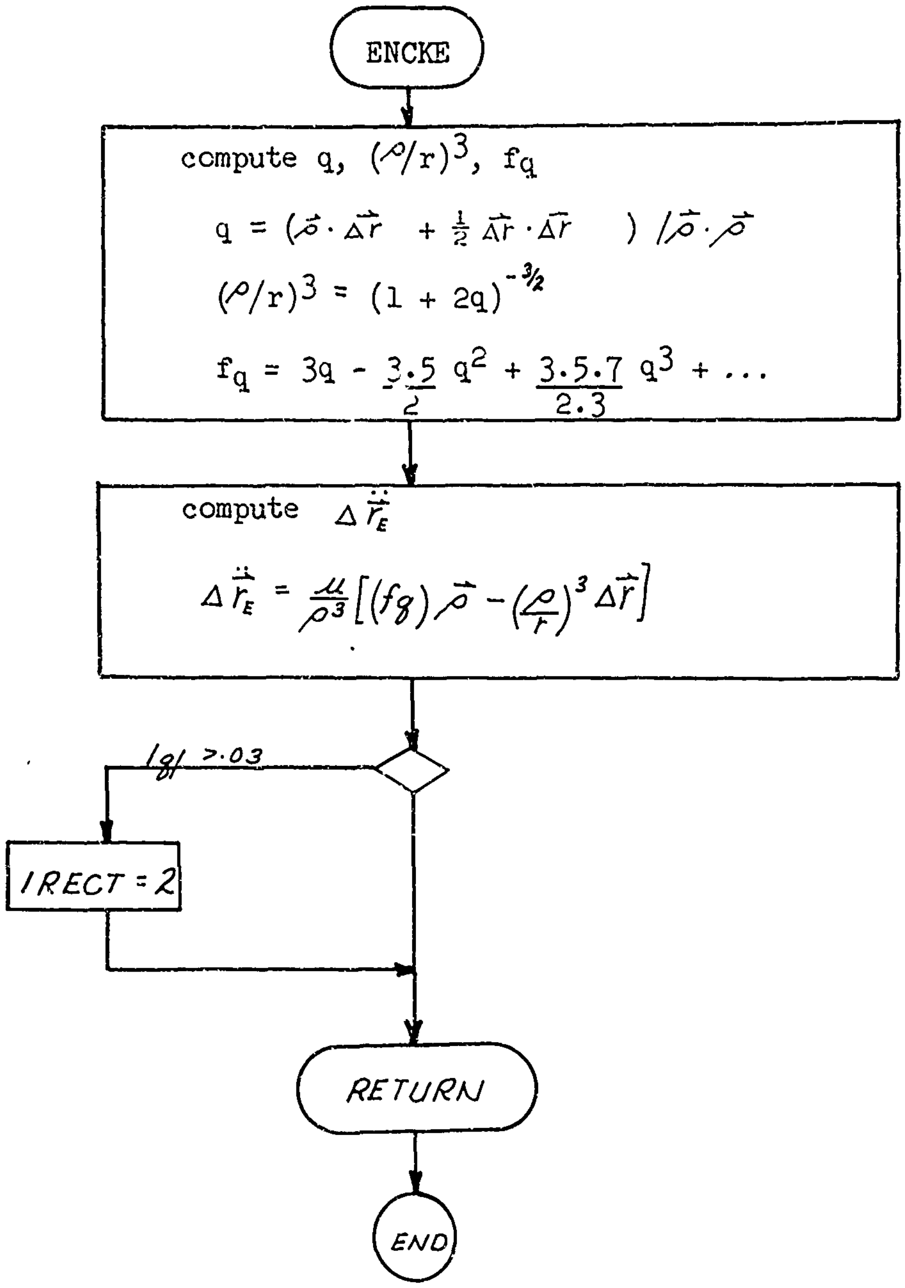
$$\frac{3\cdot 5\cdot 7\cdot 9\cdot 11\cdot 13\cdot 15}{2\cdot 3\cdot 4\cdot 5\cdot 6\cdot 7} q_{max}^7 < 10^{-8}$$

$$q_{max}^7 \approx 2.5 \times 10^{-11}$$

which corresponds roughly to

$$q_{max} = .03$$

Computational Logic:



```
      SUBROUTINE ENCKE(ACCEL,IRECT)                                     ENCKE020
C                                                                       ENCKE030
C     THIS ROUTINE COMPUTES THE ACCELERATION VECTOR EXPERIENCED BY THE  ENCKE040
C             VEHICLE DUE TO MOTION ON A TRAJECTORY OTHER THAN THE      ENCKE050
C             REFERENCE PATH. THE ENCKE SERIES IS UTILIZED TO PREVENT   ENCKE060
C             LOSS OF NUMERICAL SIGNIFICANCE IN THE REQUIRED SUBTRACT.  ENCKE070
C     THE ROUTINE ALSO GENERATES A RECTIFICATION SIGNAL WHEN DISTANCES  ENCKE080
C             FROM THE CONIC ARE EXCESSIVE. THIS STEP IS TAKEN PRIOR    ENCKE090
C             TO THE TIME THAT THE EFFECT OF SERIES TRUNCATION IS       ENCKE100
C             MEASURABLE SO AS TO ELIMINATE NEED FOR RESTARTING INTEG   ENCKE110
C                                                                       ENCKE120
C     RCONIC = POSITION VECTOR ON REFERENCE TRAJECTORY.     (1950.0)    ENCKE130
C     DELTA  = DISPLACEMENT VECTOR RELATIVE TO REF. TRAJ.   (1950.0)    ENCKE140
C                                                                       ENCKE150
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ENCKE160
C                                                                       ENCKE170
      DIMENSION  CON(1), SAT(1), SDA(1), WRK(1), STT(1)                 ENCKE180
C                                                                       ENCKE190
      COMMON DATA                                                       ENCKE200
C                                                                       ENCKE210
      EQUIVALENCE  (DATA(  1),CON), (DATA( 15),SAT), (DATA( 36),SDA)    ENCKE220
     2            ,(DATA(286),STT), (DATA(391),WRK)                     ENCKE230
C                                                                       ENCKE240
C                                                                       ENCKE250
      DIMENSION  RCONIC(3)  ,DELTA(3)  ,ACCEL(3)                        ENCKE260
C                                                                       ENCKE270
      EQUIVALENCE           (CON(  6),GM    ), (WRK( 52),DELTA )        ENCKE280
     2                     ,(WRK(117),RCONIC)                           ENCKE290
C                                                                       ENCKE300
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ENCKE310
C                                                                       ENCKE320
      R2 = DOT(RCONIC,RCONIC)                                           ENCKE330    2
      Q =(DOT(RCONIC,DELTA)+.5*DOT(DELTA,DELTA))/R2                     ENCKE340    3     4
      F = 1./(SQRT(1.+2.*Q))**3                                         ENCKE350    5
      FQ=Q*(3.+Q*(-7.5+Q*(17.5+Q*(-39.375+Q*(86.625+Q*(-187.6875+Q*     ENCKE360
     1402.1875 ))))))                                                   ENCKE370
```

STD 65-1203-1

```
     ENCK        -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      R3 = R2 *SQRT(R2)                                                ENCKE380     6
      GOF  = GM*F /R3                                                  ENCKE390
      GOFQ = GM*FQ/R3                                                  ENCKE400
      DO 1 I=1,3                                                       ENCKE410
    1 ACCEL(I) =  GOFQ*RCONIC(I) -GOF*DELTA(I)                         ENCKE420
C                                                                      ENCKE430
C                                                                      ENCKE440
C     TEST FOR RECTIFICATION * * * * * * * * * * * * * * * * * * * * * *ENCKE450
      IF(ABS(Q)-.03) 10,11,11                                          ENCKE460
   11 IRECT = 2                                                        ENCKE470
   10 CONTINUE                                                         ENCKE480
      RETURN                                                           ENCKE485
      END                                                              ENCKE490
```

STORAGE MAP

SUBROUTINE ENCKE

COMMON VARIABLES

COMMON BLOCK // ORIGIN 00001 LENGTH 00775

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DATA	00000	R	DATA	00000	R	CON	00000	R
SAT	00017	R	SDA	00043	R	STT	00435	R
WRK	00606	R	GM	00005	R	DELTA	00671	R
RCONIC	00772	R						

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
R2	00776	R	Q	00777	R	F	01000	R
FQ	01001	R	R3	01002	R	GOF	01003	R
GOFQ	01004	R						

ENTRY POINTS

ENCKE SECTION 4

SUBROUTINES CALLED

DOT SECTION 5 SQRT SECTION 6 SYSLOC SECTION 7

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1	10A	01143	10	18A	01163	11	17A	01161

```
DECK LENGTH IN OCTAL IS      00213.
```

2.3.1.2.4 Subroutine PRESS

Purpose: To compute the force exerted on a spherical satellite due to the pressure of the impinging light

Deck Name: PRES

Calling Sequence: Call PRESS (SFOR)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
O	SFOR	$\vec{F}_s$	3	Arg	Solar pressure force vector in the frame of 1950.0
I	RVEC	$\vec{r}$	3	WRK (44)	radius vector in the frame of 1950.0
I	RSUN	$\vec{r}_s$	3	WRK (98)	Position vector for the sun relative to the earth (1950.0)
I	A	A	1	SAT (2)	Cross sectional area of spherical satellite
I	R	R	1	SAT (4)	Surface reflectivity
I	RE	R_e	1	CON (1)	Equatorial radius of the earth

Subroutines Required: None

Functions Required:
AMAG (vector magnitude)
DOT (Dot product)
SPOWER (solar power)
SQRT (square root)

Approximate Deck Length: 220 (octal)

Formulation:

The solar force exerted on a spherical satellite resulting from impinging light can be obtained directly from the expression relating the solar pressure, i.e.,

$$p = \frac{P}{C} (1+R) \cos^2\alpha$$

where
- P is the power of the incident radiation
- C is the speed of light
- R is the surface reflectivity
- α is the angle between the surface normal and the impinging light.

This is accomplished by integrating the pressure over the surface and (due to symmetry) applying the result in the direction of the impinging light.

$$F_s = \frac{P}{C} (1+R) \int_A \cos^2 \alpha \, dA$$

But for the spherical satellite

$$dA = 2\pi \; S^2 \sin\alpha d\alpha$$

where
- S = radius of the satellite
- α = angle between an arbitrary radius vector to the satellite skin from its center and the vector defining the direction of the impinging light.

and the integration over the surface reduces to the integration with respect to α from 0 to $\pi/2$. The result is

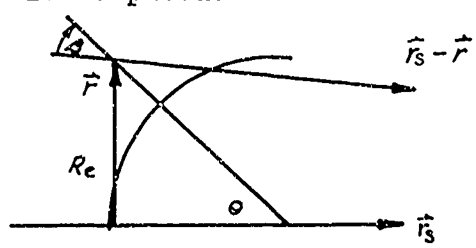
$$F_s = \frac{P}{C} (1+R) \frac{2}{3} \pi \; S^2$$

$$\vec{F}_s = -F_s \; (\vec{r}_s - \vec{r}) / |\vec{r}_s - \vec{r}|$$

This description of the problem is not, however, complete since it fails to consider the effects of any time which might be spent in the earth's shadow. This correction might be made by zeroing P for these times but for reasons of data availability in PRESS and not in SPOWER (which computes P/C) the following approach will be followed to determine if the sun is visible before the force is computed.

First, the angle between the relative sun and the radius vector is

$$\beta = \cos^{-1}\left[\frac{-\vec{r} \cdot (\vec{r}_s - \vec{r})}{r \; |\vec{r}_s - \vec{r}|}\right]$$



Then, the angle defining the limits of the cylindrical shadow can be defined as

$$\theta = \sin^{-1}\left(\frac{R_e}{r}\right)$$

and the requirement for visibility is

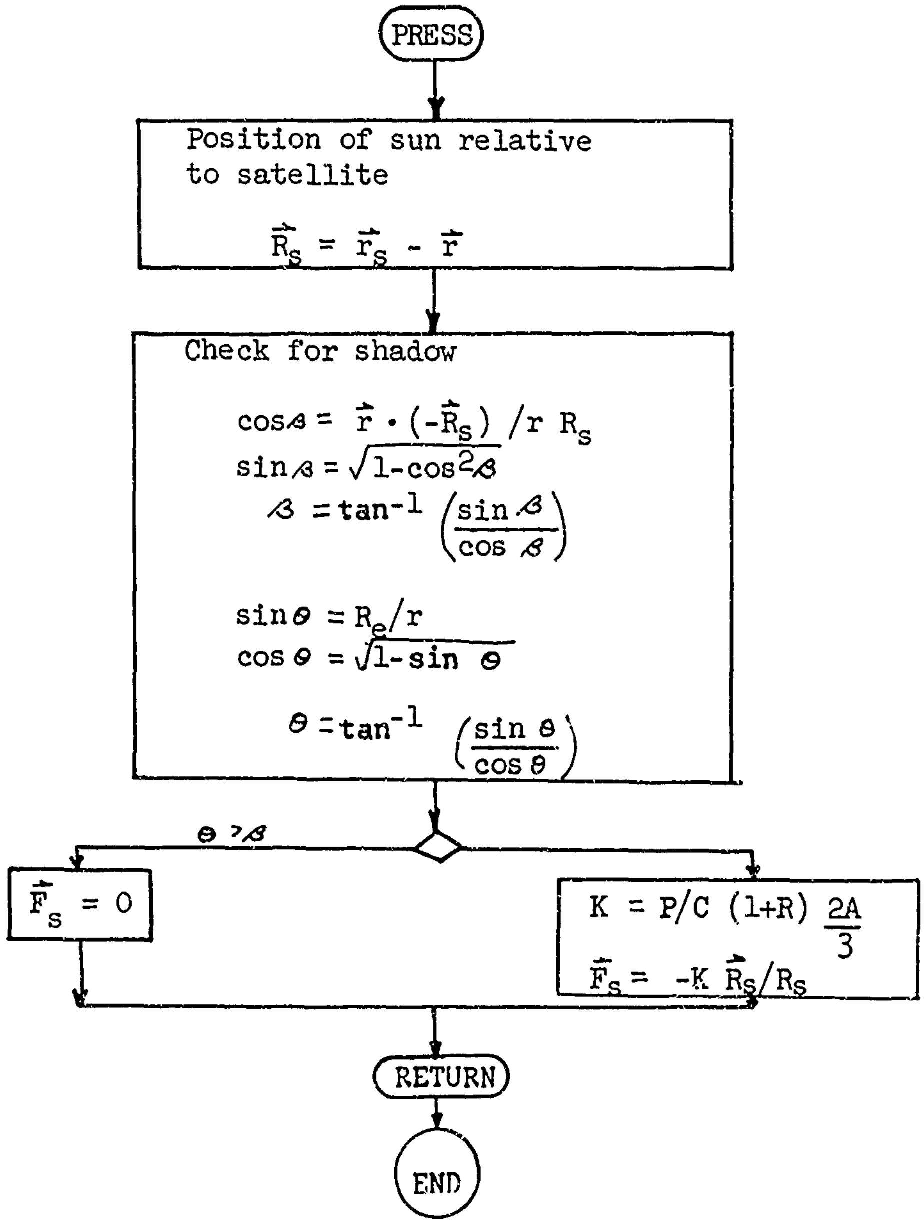
$$\beta > \theta$$

There are, of course, several assumptions which have been made which, though not completely apparent, should be noted since they introduce slight errors

1) The sun is assumed to be a point
2) The earth is assumed spherical for the purposes of defining the shadow
3) The shadow is assumed cylindrical

These assumptions are believed to be reasonable for most applications and should introduce negligible errors due to the small effect of F_s on most trajectories.

Computational Logic:



Then, the angle defining the limits of the cylindrical shadow can be defined as

$$\theta = \sin^{-1}\left(\frac{R_e}{r}\right)$$

and the requirement for visibility is

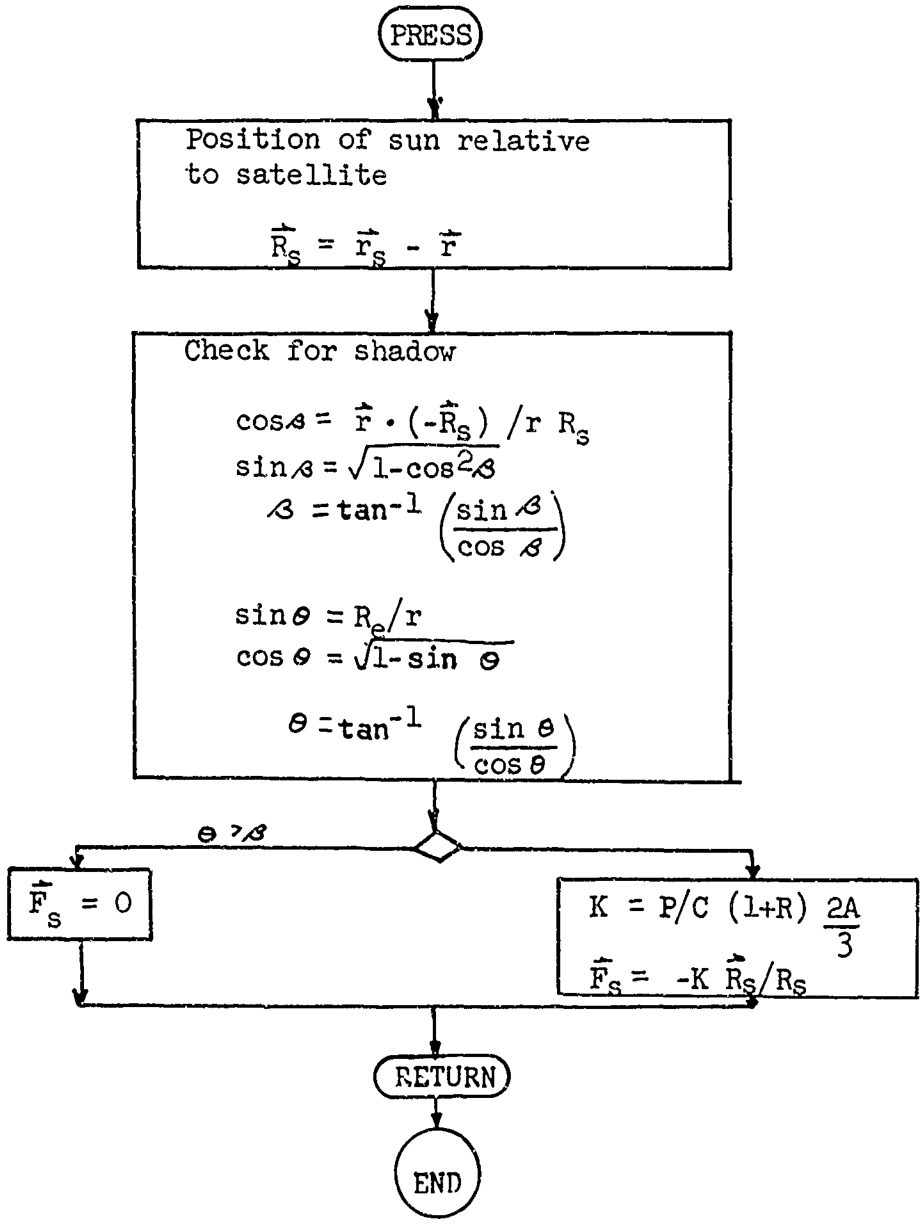
$$\beta > \theta$$

There are, of course, several assumptions which have been made which, though not completely apparent, should be noted since they introduce slight errors

1) The sun is assumed to be a point
2) The earth is assumed spherical for the purposes of defining the shadow
3) The shadow is assumed cylindrical

These assumptions are believed to be reasonable for most applications and should introduce negligible errors due to the small effect of F_s on most trajectories.

Computational Logic:



```
     PRES        -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      SUBROUTINE PRESS(SFOR)                                              PRES0020
C                                                                         PRES0030
C     THIS ROUTINE COMPUTES THE SOLAR PRESSURE FORCE EXERTED ON THE       PRES0040
C              SURFACE OF THE SAIL AND OF THE DISK .IT REQUIRES THE       PRES0050
C              CHARACTERISTICS OF THE SATELLITE FOR THIS CALCULATION      PRES0060
C              AND DATA RELATING THE ORIENTATION RELATIVE TO THE SUN.     PRES0070
C     THE FORCE BEING COMPUTED HAS THE UNITS OF NEWTONS ,LENGTH IN        PRES0080
C              KM , AREA IN METERS**2.                                    PRES0090
C     RVEC = TRUE POSITION VECTOR OF THE SATELLITE IN THE 1950.0 FRAME    PRES0100
C     RSUN = POSITION VECTOR OF THE SUNIN THE 1950.0 FRAME REL. TO EARTHPRES0110
C     A1   = CROSS SECTIONAL AREA OF THE SPHERICAL SATELLITE              PRES0120
C     R1   = REFLECTIVITY OF SATELLITE                                    PRES0130
C     SFOR = SOLAR PRESSURE FORCE EXERTED ON SATELLITE IN 1950.0 FRAME    PRES0140
C                                                                         PRES0150
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *     PRES0160
C                                                                         PRES0170
      DIMENSION  CON(1), SAT(1), SDA(1), WRK(1), STT(1)                   PRES0180
C                                                                         PRES0190
      COMMON DATA                                                         PRES0200
C                                                                         PRES0210
      EQUIVALENCE  (DATA(  1),CON), (DATA( 16),SAT), (DATA( 36),SDA)      PRES0220
     2             ,(DATA(286),STT), (DATA(391),WRK)                      PRES0230
C                                                                         PRES0240
C                                                                         PRES0250
      DIMENSION  RVEC(3)   ,RS(3)      ,RSUN(3)    ,SFOR(3)               PRES0260
C                                                                         PRES0270
      EQUIVALENCE           (CON(  1),RE     ), (SAT(  2),A1     )        PRES0280
     2  ,(SAT(  4),R1     ), (WRK( 44),RVEC   ), (WRK( 98),RSUN   )       PRES0290
C                                                                         PRES0300
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *     PRES0310
C                                                                         PRES0320
      DO 30 I=1,3                                                         PRES0330
   30 RS(I) = RSUN(I) - RVEC(I)                                           PRES0340
      RSS=AMAG(RS)                                                        PRES0350     11
      R = AMAG(RVEC)                                                      PRES0360
C                                                                         PRES0370
```

```
C        IS SPACECRAFT IN THE SHADOW. IF YES, SET FORCE = 0 AND RETURN. PRES0380
C        IF NO, CONTINUE.                                                 PRES0390  . 12
      COSBET =-DOT(RVEC,RS  )/(R*RSS)                                     PRES0400    13
      C1 = SQRT (1. - COSBET**2)                                          PRES0410    14
      ANG = ATAN2 (C1,  COSBET)                                           PRES0420    15
      C2 = RE / R                                                         PRES0430
      C3 = SQRT (1. - C2**2)                                              PRES0440    16
      ASHAD = ATAN2 (C2, C3)                                              PRES0450    17
      IF (ANG .GT. ASHAD) GO TO 70                                        PRES0460
      DO 80 I = 1,3                                                       PRES0470
   80 SFOR(I) = 0.                                                        PRES0480
      RETURN                                                              PRES0490
C                                                                         PRES0500
   70 POWER = SPOWER(RS)                                                  PRES0510    29
      SCON  = POWER * (1. + R1) * 2. * A1 / 3.                            PRES0520
      DO 10 I = 1, 3                                                      PRES0530
   10 SFOR(I) =-SCON * RS(I) / RSS                                        PRES0540
      RETURN                                                              PRES0550
      END                                                                 PRES0560
```

12/01/85

STORAGE MAP

**** PRES

SUBROUTINE PRESS

COMMON VARIABLES

COMMON BLOCK // ORIGIN 00001 LENGTH 00752

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DATA	00000	R	DATA	00000	R	CON	00000	R
SAT	00017	R	SDA	00043	R	STT	00435	R
WRK	00606	R	RE	00000	R	A1	00020	R
R1	00022	R	RVEC	00661	R	RSUN	00747	R

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
RS	00753	R						

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
RSS	00756	R	R	00757	R	COSBET	00760	R
C1	00761	R	ANG	00762	R	C2	00763	R
C3	00764	R	ASHAD	00765	R	POWER	00766	R
SCON	00767	R						

ENTRY POINTS

PRESS SECTION 4

SUBROUTINES CALLED

AMAG SECTION 5 DOT SECTION 6 SQRT SECTION 7

ATAN2 SECTION 8 SPOWER SECTION 9 SYSLOC SECTION 10

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
30	5A	01003	70	28A	01115	80	24A	01111
10	33A	01134						

DECK LENGTH IN OCTAL IS 00216.

2.3.1.2.4.1 Function SPOWER

Purpose: SPOWER computes the ratios of the solar power to the speed of light in mks units. This routine was intended to allow for motion about bodies other than the earth

Deck Name: SPOWER

Calling Sequence: SPOWER (RS)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	RS	$\vec{r}_s-\vec{r}$	3	Arg	Position of sun relative to the satellite
O	SPOWER	P/C	1	-	Ratio of solar power to the speed of light

Subroutine Required: None

Functions Required: None

Approximate Deck Length: 50 (octal)

```
      FUNCTION SPOWER(RS)                                                SPR00020
C                                                                        SPR00030
C        THIS FUNCTION COMPUTES THE SOLAR PRESSURE COEFFICIENT = P/C IN  SPR00040
C        NEWTONS / SQUARE METER                                          SPR00050
C          P = POWER OF INCIDENT LIGHT WAVE                              SPR00060
C          C = SPEED OF LIGHT                                            SPR00070
C                                                                        SPR00080
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  SPR00090
C                                                                        SPR00100
      DIMENSION  CON(1), SAT(1), SDA(1), WRK(1), STT(1)                  SPR00110
C                                                                        SPR00120
      COMMON DATA                                                        SPR00130
C                                                                        SPR00140
      EQUIVALENCE  (DATA(   1),CON), (DATA( 16),SAT), (DATA( 36),SDA)    SPR00150
     2             ,(DATA(286),STT), (DATA(391),WRK)                     SPR00160
C                                                                        SPR00170
      DIMENSION RS(3)                                                    SPR00180
C                                                                        SPR00190
      EQUIVALENCE  (CON(10),REF)                                         SPR00200
C                                                                        SPR00210
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  SPR00220
C                                                                        SPR00230
C                                                                        SPR00240
      SZERO  = .4503 E-5                                                 SPR00250
      RSS    = AMAG (RS)                                                 SPR00260     2
      SPOWER = SZERO * (REF/RSS)**2                                      SPR00270
C                                                                        SPR00280
      RETURN                                                             SPR00290
      END                                                                SPR00300
```

STØRAGE MAP

FUNCTIØN SPØWER TYPE R

COMMØN VARIABLES

COMMØN BLØCK // ØRIGIN 00001 LENGTH 00607

SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE
DATA	00000	R	DATA	00000	R	CØN	00000	R
SAT	00017	R	SDA	00043	R	STT	00435	R
WRK	00606	R	REF	00011	R			

UNDIMENSIØNED PRØGRAM VARIABLES

SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE
F.0000	00610	R	SZERØ	00611	R	RSS	00612	R

ENTRY PØINTS

SPØWER SECTIØN 4

SUBRØUTINES CALLED

AMAG SECTIØN 5 SYSLØC SECTIØN 6

EFN IFN CØRRESPØNDENCE

EFN	IFN	LØCATIØN	EFN	IFN	LØCATIØN	EFN	IFN	LØCATIØN

DECK LENGTH IN ØCTAL IS 00051.

2.3.1.2.5 Subroutine PERT

Purpose: PERT is a special purpose routine which computes the gravitational accelerations (perturbative) of the sun and moon on a geocentric satellite. The assumption is made in the development that the geocentric radial distance is small compared to the distance to the disturbing mass: thus a more exact formulation is required for the general case.

Deck Name: PERT

Calling Sequence: Call PERT (ACCEL)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	R	$\vec{r}$	3	WRK (44)	radius vector in the frame of 1950.0
I	RS	$\vec{r}_s$	3	WRK (98)	radius vector of the sun (moon) in the frame of 1950.0 relative to the Earth
	RM	$\vec{r}_m$	3	WRK (101)	
I	UM	μ_m	1	CON (8)	gravitational constant for the moon and sun
	US	μ_s	1	CON (9)	
O	ACCEL	$\Delta\ddot{\vec{r}}'$	3	Arg	perturbative acceleration vector due to gravitational acceleration of the sun and moon

Subroutines Required: None

Functions Required: AMAG (vector magnitude), FQ (Encke series function)

Approximate Deck Length: 170 (octal)

Formulation:

The acceleration experienced by a body (traveling about the Earth, for example) resulting from masses other than the central body (i.e., the sun, and moon) is the summation of terms of the following form.

$$\ddot{\vec{r}}' = -\mu\left[\frac{\vec{r}-\vec{R}}{|\vec{r}-\vec{R}|^3} - \frac{\vec{R}}{R^3}\right]$$

This equation, while correct, exhibits a very undesirable nature for $r \ll R$ since for these cases the term in the brackets is the difference of two nearly equal quantities. Therefore, since this limiting case is the case of primary interest, an alternate form is necessary if these terms in the acceleration are to be included. One such modification is based on the similarity of this perturbation equation and that for the Encke acceleration.

consider:

$$\ddot{\vec{r}}' = -\mu\left[\frac{-\vec{\rho}}{\rho^3} + \frac{\vec{R}}{R^3}\right] \qquad \rho = \vec{R} - \vec{r}$$

$$= \frac{-\mu}{R^3}\left[\vec{R} - \vec{\rho}\left(\frac{R}{\rho}\right)^3\right]$$

$$= -\frac{\mu}{R^3}\left[\vec{R} - (1-fg)\vec{\rho}\right]$$

$$= \frac{-\mu}{R^3}\left[\vec{r} + fg\vec{\rho}\right]$$

where as before, since fg has one more degree of freedom than the ratio R/ρ, g can be defined as follows

$$1 - fg = (1+2g)^{-3/2} = R^3/\rho^3$$

or

$$fg = 1 - (1+2g)^{-3/2}$$

$$= 3g - \frac{3\cdot5}{2}g^2 + \frac{3\cdot5\cdot7}{2\cdot3}g^3 - \frac{3\cdot5\cdot7\cdot9}{2\cdot3\cdot4}g^4 + \ldots$$

also

$$\frac{\rho^2}{R^2} = 1 + 2g$$

$$q = \frac{\frac{\rho^2}{R^2} - 1}{2} = \frac{1}{2R^2}\left(\rho^2 - R^2\right)$$
$$= -\frac{1}{2R^2}\left(R^2 - \rho^2\right)$$

but

$$\vec{\rho} = \vec{R} - \vec{r}$$
$$= (X - x)\,\hat{x} \cdots$$

therefore

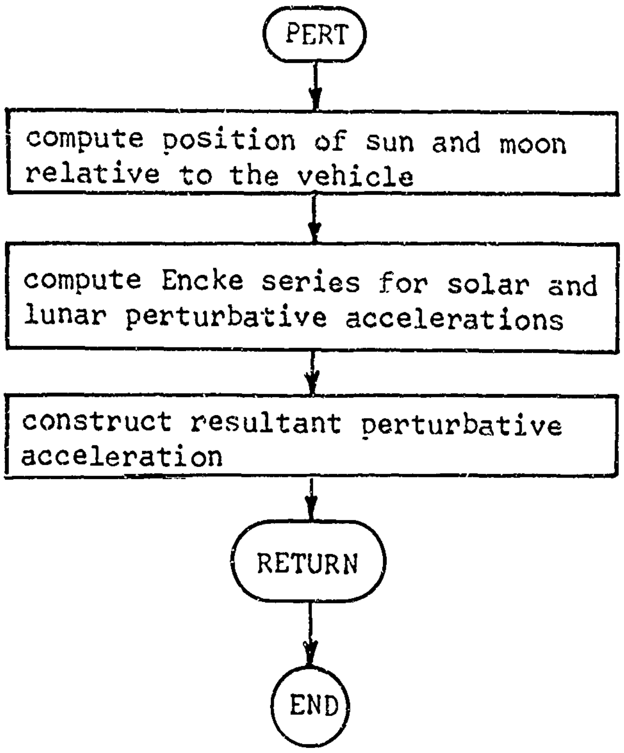
$$q = -\frac{1}{2R^2}\left(X^2 + Y^2 + Z^2 - (X - x)^2 - (Y - y)^2 - (Z - z)^2\right)$$

$$= -\frac{1}{2R^2}\left(2(Xx + Yy + Zz) - (x^2 + y^2 + z^2)\right)$$

$$= -\frac{1}{R^2}\left(\vec{R}\cdot\vec{r} - \vec{r}\cdot\vec{r}\right)$$

$$= \frac{-\vec{R}\cdot\vec{r} + (\vec{r}\cdot\vec{r})/2}{R^2}$$

Computational Logic:



```
      SUBROUTINE PERT(ACCEL)                                               PERT0020
C                                                                          PERT0030
C     THIS ROUTINE COMPUTES THE PURTURBATIVE ACCELERATIONS EXPERIENCED    PERT0040
C             BY A SATELLITE DUE TO THE SUN AND MOON . THIS                PERT0050
C             ACCELERATION IS COMPUTED UTILIZING THE ENCKE SERIES          PERT0060
C             RATHER THAN BY UTILIZING THE DIFFERENCES IN NEARLY EQUAL     PERT0070
C             TERMS OF THE TYPE   ( 1/R)**3 .                              PERT0080
C                                                                          PERT0090
C                                                                          PERT0100
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    PERT0110
C                                                                          PERT0120
      DIMENSION  CON(1), SAT(1), SDA(1), WRK(1), STT(1)                    PERT0130
C                                                                          PERT0140
      COMMON DATA                                                          PERT0150
C                                                                          PERT0160
      EQUIVALENCE  (DATA(  1),CON), (DATA( 16),SAT), (DATA( 36),SDA)       PERT0170
     2             ,(DATA(286),STT), (DATA(391),WRK)                       PERT0180
C                                                                          PERT0190
C                                                                          PERT0200
      DIMENSION  R(3)       ,RS(3)      ,RM(3)      ,RHOS(3)    ,RHOM(3)   PERT0210
     2,ACCLS(3) ,ACCLM(3)   ,ACCEL(3)   ,XRS(3)     ,XRM(3)                PERT0220
C                                                                          PERT0230
      EQUIVALENCE            (CON(  8),UM     ), (CON(  9),US     )        PERT0240
     2  ,(WRK( 44),R      ), (WRK( 98),RS     ), (WRK(101),RM     )        PERT0250
C                                                                          PERT0260
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    PERT0270
C                                                                          PERT0280
      DO 1 I= 1,3                                                          PERT0290
      RHOS(I) =-R(I) + RS(I)                                               PERT0300
    1 RHOM(I) =-R(I) + RM(I)                                               PERT0310
      RSS3 = (AMAG(RS))**3                                                 PERT0320    14
      RMM3 = (AMAG(RM))**3                                                 PERT0330    15
      DO 20 J = 1,3                                                        PERT0340
      XRS(J) = -RS(J)                                                      PERT0350
   20 XRM(J) = -RM(J)                                                      PERT0360
      FQS  = FQ(XRS,R)                                                     PERT0370    26
```

```
    PERTA       -  EFN   SOURCE STATEMENT  -  IFN(S)

   FQM  = FQ(XRM,R)                                              PERT0380     27
   DO 2 I=1,3                                                    PERT0390
   ACCLS(I) = -US*( R(I) + RHOS(I) * FQS ) / RSS3                PERT0400
   ACCLM(I) = -UM*( R(I) + RHOM(I) * FQM ) / RMM3                PERT0410
 2 ACCEL(I) = ACCLS(I) + ACCLM(I)                                PERT0420
   RETURN                                                        PERT0430
   END                                                           PERT0440
```

```
                                    STORAGE MAP

                               SUBROUTINE  PERT

                                 COMMON   VARIABLES

                COMMON BLOCK     //            ORIGIN      00001        LENGTH     00755

SYMBOL      LOCATION     TYPE     SYMBOL    LOCATION    TYPE    SYMBOL    LOCATION    TYPE
DATA         00000        R       DATA       00000       R      CON        00000       R
SAT          00017        R       SDA        00043       R      STT        00435       R
WRK          00606        R       UM         00007       R      US         00010       R
R            00661        R       RS         00747       R      RM         00752       R

                            DIMENSIONED PROGRAM VARIABLES

SYMBOL      LOCATION     TYPE     SYMBOL    LOCATION    TYPE    SYMBOL    LOCATION    TYPE
RHOS         00756        R       RHOM       00761       R      ACCLS      00764       R
ACCLM        00767        R       XRS        00772       R      XRM        00775       R

                           UNDIMENSIONED PROGRAM VARIABLES

SYMBOL      LOCATION     TYPE     SYMBOL    LOCATION    TYPE    SYMBOL    LOCATION    TYPE
RSS3         01000        R       RMM3       01001       R      FQS        01002       R
FQM          01003        R

                                    ENTRY POINTS

    PERT        SECTION     4

                                  SUBROUTINES CALLED

    AMAG        SECTION     5        FQ        SECTION    6       SYSLOC     SECTION     7
```

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1	8A	G1C16	20	21A	C1C52	2	37A	C1115

DECK LENGTH IN OCTAL IS C017C.

2.3.1.2.5.1 Function FQ

Purpose: to evaluate the Encke series utilized in constructing solar and lunar gravitational accelerations of close geocentric satellites.

Deck Name: FQ

Calling Sequence: FQ (RN, R)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	RN	$\vec{R}$	3	Arg	radius vector for the sun (or moon) in the frame of 1950.0
I	R	$\vec{r}$	3	Arg	radius vector for the vehicle in the frame of 1950.0
0	FQ	f_q	1	-	numerical value of Encke series

Subroutines Required: None

Functions Required: None

Approximate Deck Length: 120 (octal)

```
    FQQ          -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      FUNCTION FQ(RN,R)                                                  FQ**0020
C     THIS ROUTINE EVALUATES THE ENCKE SERIES TO BE UTILIZED IN THE      FQ**0030
C             EVALUATION OF THE EFFECTS OF PERTURBATIONS PRODUCED        FQ**0040
C             EITHER BY EXTRA-TERRESTRIAL MASSES OR BY MOTION ON OTHER   FQ**0050
C             THAN THE CONIC REFERENCE TRAJECTORY                        FQ**0060
C                                                                        FQ**0070
C                                                                        FQ**0080
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  FQ**0090
C                                                                        FQ**0100
      DIMENSION  RN(3),R(3)                                              FQ**0110
C                                                                        FQ**0120
      Q =( DOT(RN,R ) +.5*DOT(R,R))/ DOT(RN,RN)                          FQ**0130    2    3    4
      FQ= Q*(3.+Q*(-7.5+Q*(17.5 +Q*(-39.375+Q*(86.625+Q*(-187.6875))))))FQ**0140
      RETURN                                                             FQ**0150
      END                                                                FQ**0160
```

```
FS305                                              11/23/85              PAGE  1C4
      FQQ                           STORAGE  MAP

                          FUNCTION    FQ          TYPE     R

                              UNDIMENSIONED PROGRAM VARIABLES

SYMBOL      LOCATION     TYPE     SYMBOL      LOCATION     TYPE     SYMBOL      LOCATION     TYPE
F.000C       C0001        R       Q            00002        R

                                   ENTRY POINTS

     FQ          SECTION      3

                                   SUBROUTINES CALLED

     DOT         SECTION      4          SYSLOC      SECTION     5

                              EFN    IFN   CORRESPONDENCE

EFN         IFN         LOCATION    EFN         IFN         LOCATION    EFN         IFN         LOCATION

  DECK LENGTH IN OCTAL IS      00121.
```

2.3.1.2.6 Subroutine EPHEM

Purpose: EPHEM is designed to compute approximate position vectors for the sun and moon from two-body equations of motion and tabulated position and velocity information obtained from the Data Input Group.

Deck Name: EPHEM

Calling Sequence: Call EPHEM

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	TW TF	t	1 1	WRK (50) WRK (51)	The whole and functional part of a mean solar day defining the desired epoch for $\vec{r}_s$ and $\vec{r}_m$
O	RS RM	$\vec{r}_s$ $\vec{r}_m$	3 3	WRK (98) WRK (101)	Solar and lunar position vectors in the coordinate frame of 1950.0
I	GME GMM GMS	μ μ_m μ_s	1 1 1	CON (6) CON (8) CON (9)	gravitational constants for the Earth, the moon and the sun (in Km^3/sec^2)
I	EPHAM	$\vec{r}_i$, $\vec{v}_i$	339	EPHOM	Tabulated position and velocity vectors for the sun and moon for the time period 1965 - 1975
I	EPHM	$\vec{r}_{i_{☾}}$, $\vec{v}_{i_{☾}}$	271	EPHAM (4)	Subarray containing the lunar data
I	EPHS	$\vec{r}_{i_{\odot}}$, $\vec{v}_{i_{\odot}}$	66	EPHAM (274)	Subarray containing the solar data

Subroutines Required: CONIC (conic section routine)

Functions Required: None

Approximate Deck Length: 456 (octal)

Formulation:

The equations of two-body motion (i.e., the Earth-sun and the Earth-moon systems) relative to the center of mass are

$$\ddot{\vec{r}}_1 = -G m_2 \left(\frac{m_2}{m_1 + m_2}\right)^2 \frac{\vec{r}_1}{r_1^3}$$

$$\ddot{\vec{r}}_2 = -G m_1 \left(\frac{m_1}{m_1 + m_2}\right)^2 \frac{\vec{r}_2}{r_2^3}$$

which are immediately recognizable as the equations of conic motion. This fact will be utilized in the following paragraphs to approximate the true ephemeris of the moon and sun relative to the Earth.

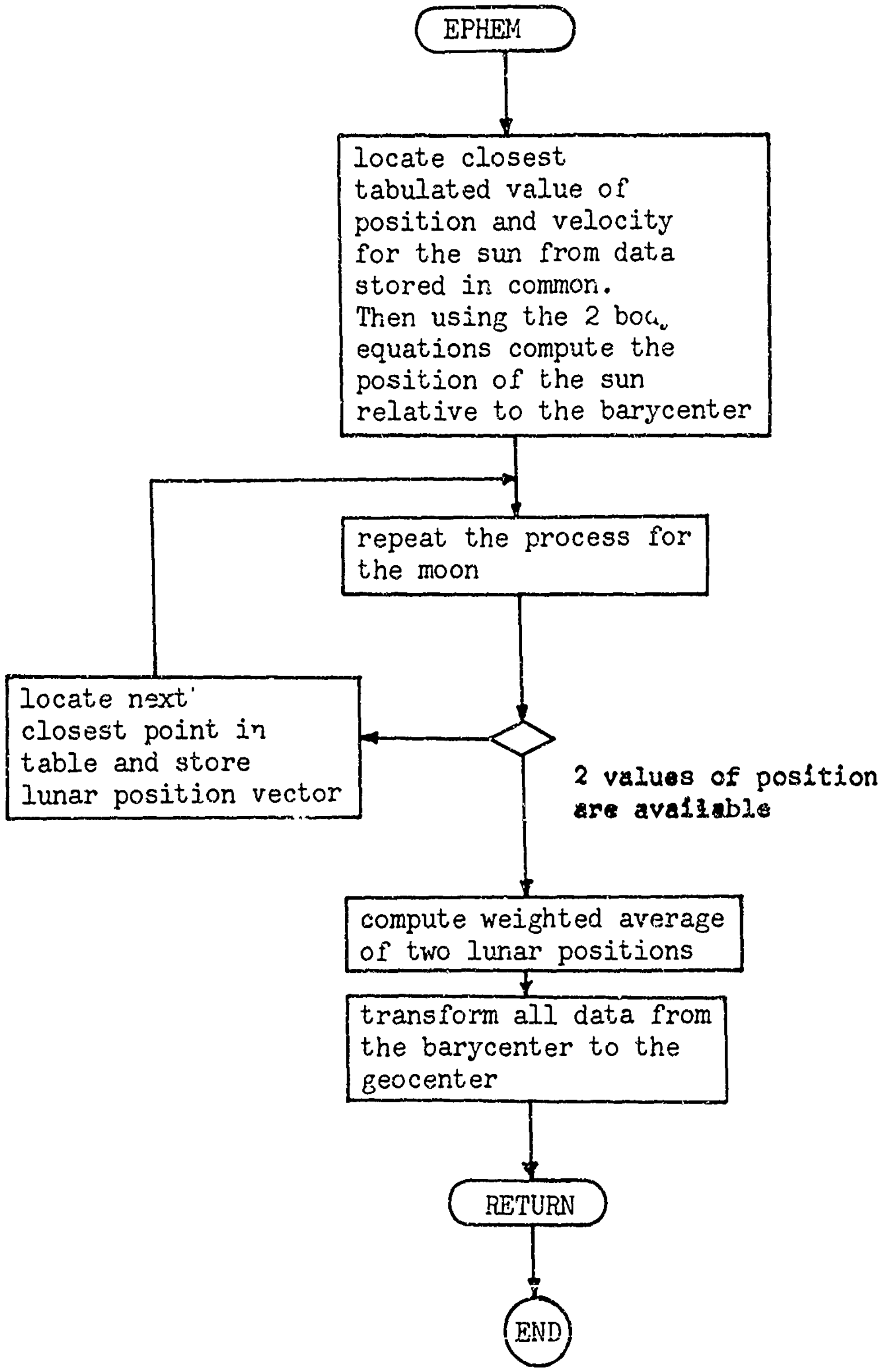
Consider a table of values for $\vec{r}_0$ and $\vec{v}_0$ at regular intervals of approximately 1 year from which the conic describing the motion of the sun (relative to barycenter) can be computed. The problem is simply to locate the closest tabulated point (no more than about 6 months), define the time over which the body has moved, and compute $\vec{r}$ and $\vec{v}$ utilizing the conic section routine developed for the reference trajectory. Since this conic is altered but a very little by planetary perturbations, the result is of moderate precision.

Now consider the problem of the moon's motion when the sun produces a significant perturbation. First, it is obvious that the time interval between values of $\vec{r}_0$ and $\vec{v}_0$ must be reduced so that the effect of the perturbations can be kept small (for the present case Δt = 3 months). Second, it is apparent that the results obtained with the technique would be more accurate if they could include some of the effects predicted if the next-to-closest point in the table were utilized for extrapolation purposes. This correction was included by weighting two separate computations of the position vector (in a manner which resulted in even weighting for a date midway from either point in the table and a weight of 1 for the term designed to include the effects of the closest point if the time was equal to a tabular time) and averaging the results.

The final step in both cases was to translate the origin from the barycenter to the geocenter.

Results prepared by this routine have been checked against a true ephemeris to compute the errors inherent in the process. This analysis showed that the solar position data were completely satisfactory (from the standpoint of accuracy) for the computation of both the solar radiation force and the

solar gravitational acceleration in that the noise in the data would normally produce errors in the total perturbating acceleration (from MOTION) which were beyond the 8-digit limitations of the machine. The same is not quite true (for near-Earth satellites) of the lunar ephemeris even though the correction cycle is employed. However, it is noted, in defense of its use, that the second integral of the total resultant differential acceleration ($\Delta\vec{r}$) is in itself a correction to the conic position vector ($\vec{r}_c$). Therefore, the noise which is not lost when summing the perturbative terms in MOTION is generally lost when adding $\Delta\vec{r}$ to $\vec{r}_c$.


EPHEM
locate closest tabulated value of position and velocity for the sun from data stored in common. Then using the 2 body equations compute the position of the sun relative to the barycenter
repeat the process for the moon
locate next closest point in table and store lunar position vector
2 values of position are available
compute weighted average of two lunar positions
transform all data from the barycenter to the geocenter
RETURN
END

```
      SUBROUTINE EPHEM                                                  EPHM0020
C                                                                       EPHM0030
C     THIS ROUTINE IS DESIGNED TO PROVIDE AN APPROXIMATE EPHEMERIS      EPHM0040
C             FOR THE SUN AND MOON SO THAT GRAVITATIONAL FORCES AND     EPHM0050
C             SOLAR PRESSURE FORCES CAN BE INCLUDED IN THE INTEGRATED   EPHM0060
C             TRAJECTORY. THE APPROXIMATION INVOLVED IS THAT THE        EPHM0070
C             ORBITS OF THESE BODIES ARE ELLIPTIC WITH INITIAL          EPHM0080
C             CONDITIONS DEFINED BY THE TIME(T) AND STORED IN           EPHM0090
C             THE INPUT ROUTINE .                                       EPHM0100
C     TW,TF = TIME IN DAYS FROM 1950.0                                  EPHM0110
C     RS,RM = POSITION VECTORS FOR SUN AND MOON IN FRAME OF 1950.0 (KM) EPHM0120
C                                                                       EPHM0130
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * EPHM0140
C                                                                       EPHM0150
      DIMENSION  CON(1), SAT(1), SDA(1), WRK(1), STT(1)                 EPHM0160
C                                                                       EPHM0170
      COMMON DATA                                                       EPHM0180
C                                                                       EPHM0190
      EQUIVALENCE  (DATA(  1),CON), (DATA( 16),SAT), (DATA( 36),SDA)    EPHM0200
     2             ,(DATA(286),STT), (DATA(391),WRK)                    EPHM0210
C                                                                       EPHM0220
      DIMENSION    RS(3)      ,RM(3)      ,RO(3)      ,VO(3)            EPHM0230
     2,EPHM(270)  ,EPHS(66)   ,VS(3)      ,RM1(3)                       EPHM0240
C                                                                       EPHM0250
      COMMON  /EPHOM/EPHAM(339)                                         EPHM0260
C                                                                       EPHM0270
C                                                                       EPHM0280
      EQUIVALENCE            (CON(  6),GME    ), (CON(  8),GMM    )     EPHM0290
     2                      ,(CON(  9),GMS    )                         EPHM0300
      EQUIVALENCE            (WRK( 50),TW     ), (WRK( 51),TF     )     EPHM0310
      EQUIVALENCE            (WRK( 98),RS     ), (WRK(101),RM     )     EPHM0320
      EQUIVALENCE            (EPHAM(4),EPHM   ), (EPHAM(274),EPHS)      EPHM0330
C                                                                       EPHM0340
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   EPHM0350
C                                                                       EPHM0360
C     SOLAR EPHEMERIS * * * * * * * * * * * * * * * * * * * * * * * * * EPHM0370
```

```
      TOS = EPHAM(1)                                              EPHM0380
      DTS = EPHAM(2)                                              EPHM0390
      TS=(TW-TOS) + TF                                            EPHM0400
      N = TS/DTS +.5                                              EPHM0410
      IF(N) 10,10,11                                              EPHM0420
   10 N = 0                                                       EPHM0430
      GO TO 12                                                    EPHM0440
   11 IF(N-9) 12,12,13                                            EPHM0450
   13 N = 9                                                       EPHM0460
   12 J = 6*N                                                     EPHM0470
      DO 1 I=1,3                                                  EPHM0480
      K = J+I                                                     EPHM0490
      KK= K+3                                                     EPHM0500
      RO(I) = EPHS(K )                                            EPHM0510
    1 VO(I) = EPHS(KK)                                            EPHM0520
      SAVE = GME                                                  EPHM0530
      GME = 1.3271455E11                                          EPHM0540
      X =N                                                        EPHM0550
      T =(TS - X*DTS)*86400.                                      EPHM0560
      DT= T                                                       EPHM0570
      CALL CONIC( RO,VO,T,O,DT,RS,VS )                            EPHM0580
C                                                                 EPHM0590
C                                                                 EPHM0600
C     LUNAR POSITION * * * * * * * * * * * * * * * * * * * * * * * *  EPHM0610   23
      TOS = EPHAM(1)                                              EPHM0620
      DTS = EPHAM(3)                                              EPHM0630
      TS  =(TW -TOS) + TF                                         EPHM0640
      IJK = 1                                                     EPHM0650
      N = TS/DTS +.5                                              EPHM0660
   32 IF(N) 20,20,21                                              EPHM0670
   20 N = 0                                                       EPHM0680
      GO TO 22                                                    EPHM0690
   21 IF(N-44) 22,22,23                                           EPHM0700
   23 N = 44                                                      EPHM0710
   22 J = 6*N                                                     EPHM0720
      DO 2 I=1,3                                                  EPHM0730
      K = J+I                                                     EPHM0740
```

```
      KK= K+3                                                        EPHM0750
      RO(I) = EPHM(K )                                               EPHM0760
    2 VO(I) = EPHM(KK)                                               EPHM0770
      GME = 388977.82                                                EPHM0780
      X = N                                                          EPHM0790
      T =(TS - X*DTS)*86400.                                         EPHM0800
      DT= T                                                          EPHM0810
      CALL CONIC( RO,VO,T,0,DT,RM,VS )                               EPHM0820    47
      IF(IJK-1) 30,30,40                                             EPHM0830
   30 IJK = 2                                                        EPHM0840
      NY = TS/DTS                                                    EPHM0850
      IF( N-NY ) 86,86,87                                            EPHM0860
   86 N = N+1                                                        EPHM0870
      GO TO 88                                                       EPHM0880
   87 N = N-1                                                        EPHM0890
   88 DO 31 I=1,3                                                    EPHM0900
   31 RM1(I) = RM(I)                                                 EPHM0910
      GO TO 32                                                       EPHM0920
   40 T = T/86400.                                                   EPHM0930
      IF(T) 98,98,99                                                 EPHM0940
   98 T = DTS + T                                                    EPHM0950
   99 XX = T/DTS                                                     EPHM0960
      YY = 1. - XX                                                   EPHM0970
      DO 41 I=1,3                                                    EPHM0980
   41 RM(I) = XX*RM(I) + YY*RM1(I)                                   EPHM0990
C                                                                    EPHM1000
C                                                                    EPHM1010
C     SHIFT ORIGIN TO EARTH CENTER * * * * * * * * * * * * * * * * * *EPHM1020
      DO 68 I=1,3                                                    EPHM1030
      RS(I) = RS(I) - .012150447 * RM(I)                             EPHM1040
      RM(I) = 1.0121505 * RM(I)                                      EPHM1050
   68 CONTINUE                                                       EPHM1060
      GME = SAVE                                                     EPHM1070
      RETURN                                                         EPHM1080
      END                                                            EPHM1090
```

STORAGE MAP

SUBROUTINE EPHEM

COMMON VARIABLES

COMMON BLOCK // ORIGIN 00001 LENGTH 00755

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DATA	00000	R	DATA	00000	R	CON	00000	R
SAT	00017	R	SDA	00043	R	STT	00435	R
WRK	00606	R	GME	00005	R	GMM	00007	R
GMS	00010	R	TW	00667	R	TF	00670	R
RS	00747	R	RM	00752	R			

COMMON BLOCK EPHOM ORIGIN 00756 LENGTH 00523

EPHAM	00000	R	EPHAM	00000	R	EPHM	00003	R
EPHS	00421	R						

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
R0	01501	R	V0	01504	R	VS	01507	R
RM1	01512	R						

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
TOS	01515	R	DTS	01516	R	TS	01517	R
N	01520	I	J	01521	I	I	01522	I
K	01523	I	KK	01524	I	SAVE	01525	R
X	01526	R	T	01527	R	DT	01530	R
IJK	01531	I	NY	01532	I	XX	01533	R
YY	01534	R						

ENTRY POINTS

EPHEM SECTION 6

SUBROUTINES CALLED

CONIC	SECTION 7	E.1	SECTION 8	E.2	SECTION 9		
E.3	SECTION 10	E.4	SECTION 11	CC.1	SECTION 12		
CC.2	SECTION 13	CC.3	SECTION 14	CC.4	SECTION 15		
SYSLOC	SECTION 16						

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
10	4A	01602	11	6A	01604	12	9A	01612
13	8A	01610	1	18A	01633	32	26A	01723
20	28A	01726	21	30A	01730	22	33A	01736
23	32A	01734	2	42A	01757	30	50A	02027
40	67A	02061	35	54A	02044	87	56A	02050
88	57A	02053	31	61A	02054	98	69A	02067
99	70A	02072	41	74A	02101	68	88A	02123

DECK LENGTH IN OCTAL IS 00446.

2.3.1.3 THE INTEGRATION GROUP

The differential acceleration vector (relative to the central force acceleration describing the reference ellipse) is integrated to define both the differential position and velocity vectors with a Gauss-Jackson backwards difference formulation. This solution is accomplished by mechanizing several separate routines.

INGRAT	The driver routine
HSIZE	Dynamically varies the step size for the integration to maximize accuracy and efficiency
START	A fourth order Runge-Kutta integration routine utilized to establish the required difference table.
DIFTAB	Evaluates the leading diagonal in the sum-difference tables
INTEG	Gauss-Jackson backwards difference uncorrected integration

The various portions of this group are in themselves general purpose routines. However, the group as a whole has been designed as a special purpose routine for the differential corrections program. This design philosophy can be understood when the requirements for the routines operation, the nature of the acceleration being integrated, and the requirements for communication between the calling program and INGRAT are analyzed. This approach also affords the maximum efficiency from this integration concept by eliminating checks for alternate stops, etc.

Subroutine INGRAT

Purpose: INGRAT serves as the driver for the integration of the equations of motion employed in this program and is intended to function in the mode of a Gauss-Jackson integration. Complete logic is included for starting the integration and/or varying the step size and restarting the process.

Deck Name: INGRT

Calling Sequence: Call INGRAT (ISTART)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I/O	ISTART	-	1	ARG	A fixed point index utilized to define the mode of operation for INGRAT. ISTART = 1 for Runge-Kutta starter; = 2 for Gauss-Jackson continuation
I	CONV1	$\frac{sec}{day}$	1	CON (11)	Conversion from mean solar days to seconds
I	RCO	$\vec{r}_c$	3	WRK (28)	Position and velocity vectors for the conic reference trajectory at the time of the most recent rectification (Km,Km/sec) in the frame of 1950.0
	VCO	$\vec{v}_c$	3	WRK (31)	
I/O	RVEC	$\vec{r}$	3	WRK (44)	The position and velocity vectors in the mean equator of date frame of 1950.0 (Km, Km/sec)
	VVEC	$\vec{v}$	3	WRK (47)	
I/O	TW	t	1	WRK (50)	The whole and fractional number of days elapsed since the reference date 1950.0 (JD 2433282.423) for the present epoch
	TF		1	WRK (51)	

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I I/O I/O	RDD RD R	$\Delta\ddot{\vec{r}}$ $\Delta\dot{\vec{r}}$ $\Delta\vec{r}$	3 3 3	WRK (58) WRK (55) WRK (52)	The acceleration, velocity, and position vectors defining the true motion relative to the reference ellipse in the frame of 1950.0 (Km/sec^2, Km/sec, Km)
I	TSTOPW TSTOPF	t_{stop}	1 1	WRK (61) WRK (62)	The whole and fractional number of days since 1950.0 for the data
I/O	H	h	1	WRK (68)	The step size in seconds for the numerical integration
I	RS RM	$\vec{r}_s$ $\vec{r}_m$	3 3	WRK (98) WRK (101)	The position vectors for the sun and moon in the form of 1950.0 (Km)
I/O	TIME	$t-t_c$	1	WRK (104)	The time interval in seconds since the reference trajectory was last rectified

Subroutines Required: CONIC (conic reference trajectory)
START (Runge-Kutta starter)
DIFTAB (difference table)
INTEG (Gauss-Jackson integration)
HSIZE (step size control)

Functions Required: AMAG (vector magnitude)

Approximate Deck Length: 501 (octal)

Discussion:

INGRAT is a special-purpose driver routine for a step-by-step numerical integration of the equations of motion as developed for this program. This integration is accomplished by mechanizing a step size check routine and either a fourth-order Runge-Kutta starter or a Gauss-Jackson (double-sum) backwards difference (through sixth difference) continuation routine.

INGRAT accomplishes its objective by continuously monitoring the time interval from the present epoch to the time at which data will be available and defining the information required during the checks performed within HSIZE to assure that an optimum step size can be utilized to arrive at the target epoch exactly in the fewest number of Gauss-Jackson steps possible. In addition, INGRAT serves as the means by which data generated in START (the fourth order Runge-Kutta starter) are stored and differenced to set up the solution by the Gauss-Jackson routine.

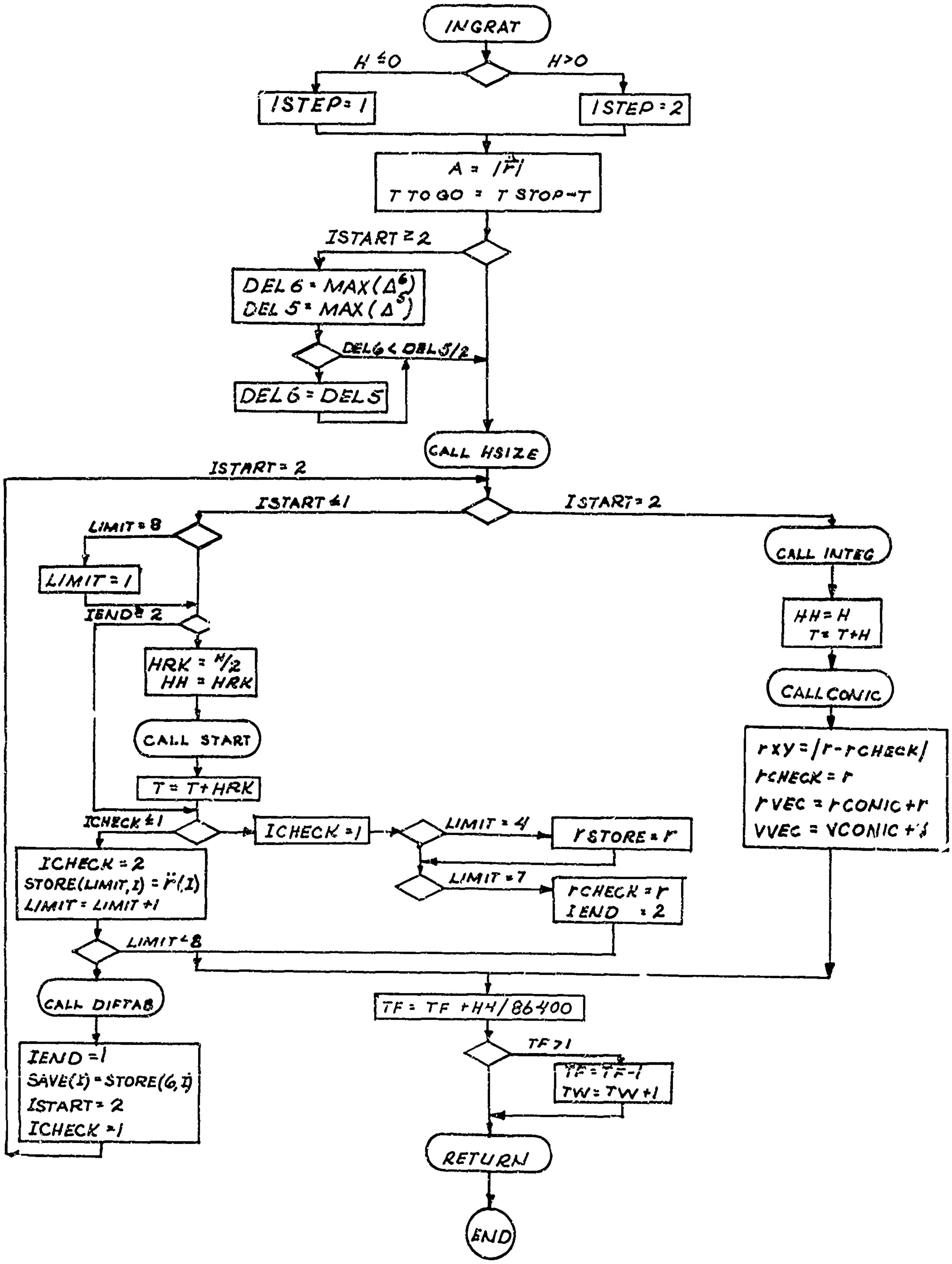
Several steps are taken to assure that the solution proceeds in an accurate and efficient manner. However, no corrector cycle has been employed either in the starting process or in the continuation process. This step is felt justifiable since:

1. The equations being integrated represent the displacement from and velocity relative to the reference ellipse (Thus, even moderate errors will not generally be realized when the true position and velocity vectors are computed by adding the results of the integration to the conic position and velocity.).

2. The Gauss-Jackson formulation has demonstrated superior performance capabilities when compared to the Runge-Kutta formulation in the integration of $\ddot{X} = -KX$ for similar step sizes without employing the corrector. (For this reason, the Runge-Kutta method [which is employed to start the Gauss-Jackson integration] will utilize a step which is one-half of that to be employed in the Gauss-Jackson method.)

3. The step size itself will be controlled in such a manner that the predictor formula alone will provide the required accuracy by monitoring the contribution of the last term carried in the integration series to the integral (this procedure will cause the step size for this routine to be less than that for a predictor-corrector formulation; however, sample calculations have shown that the slight improvement in accuracy [about one digit] and step size [a factor of about 2]) does not justify this procedure for this application.

Computational Logic:



```
      SUBROUTINE INGRAT(ISTART)
C
C     INGRAT IS THE DRIVER ROUTINE FOR THE INTERGRATION PACKAGE . THE
C             FIRST OPERATION TO BE PERFORMED WILL BE TO SET UP THE
C             STEP SIZE , THEN A CHECK WILL BE MADE TO SEE IF THE
C             INTEGRATION IS STARTING OR CONTINUING . THE PROGRAM WILL
C             THEN BRANCH TO A 4-TH ORDER RUNGE-KUTTA OR TO A 5-TH
C             DIFFERENCE-DOUBLE SUM INTEGRATION ,STEP THE FUNCTION
C             AND RETURN TO THE MAIN PROGRAM.
C     RVEC,VVEC = RADIUS AND VELOCITY VECTORS (1950.0)
C     RDD,RD,R  = ACCELERATION,RATE,AND POSITION VECTORS FOR PERTURB
C     TIME      = SECONDS FROM LAST RECTIFICATION
C     H         = STEP SIZE (ONE WILL BE SELECTED UNLESS H IS GIVEN)
C     TW,TF     = WHOLE AND FRACTIONAL PARTS OF THE PRESENT DAY
C     TSTOPW(F) = WHOLE AND FRACTIONAL PARTS OF DAY FOR TERMINATION
C     ISTART     =1 TO START WITH RUNGE KUTTA ** =2 FOR GAUSS
C             CONTINUATION ROUTINE
C
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION  CON(1), SAT(1), SDA(1), WRK(1), STT(1)
C
      COMMON DATA
C
      EQUIVALENCE  (DATA(   1),CON), (DATA( 16),SAT), (DATA( 36),SDA)
     2             ,(DATA(286),STT), (DATA(391),WRK)
C
C
      DIMENSION  RCO(3)    ,VCO(3)    ,RVEC(3)   ,VVEC(3)   ,R(3)
     2,RD(3)     ,RDD(3)    ,COEF(8,3) ,A(4)                 ,RSTORE(3)
     3,VSTORE(3),STORE(7,3),RCONIC(3) ,VCONIC(3) ,SAVF(3)    ,RCHECK(3)
     4,RXX(3)
C
      EQUIVALENCE          (CON( 11),CONV1 )
      EQUIVALENCE          (WRK( 28),RCO    ), (WRK( 31),VCO    )
     2  ,(WRK( 44),RVEC  ), (WRK( 47),VVEC   ), (WRK( 52),R      )
     3  ,(WRK( 55),RD    ), (WRK( 58),RDD    ), (WRK( 61),TSTOPW)
```

```
      4  ,(WRK( 62),TSTOPF), (WRK( 68),H      ), (WRK( 71),COEF  )      INGR0380
      5  ,(WRK( 50),TW    ), (WRK( 51),TF     ), (WRK(104),TIME  )      INGR0390
      6  ,(WRK(123),HRK   ), (WRK(130),A      ), (WRK(134),NSTEPS)      INGR0400
C                                                                       INGR0410
      DATA  ICHECK,IEND,L,LIMIT/1,1,1,1/
C                                                                       INGR0420
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * INGR0430
C                                                                       INGR0440
      IF(H) 100,100,200                                                 INGR0450
  100 ISTEP = 1                                                         INGR0460
      GO TO 101                                                         INGR0470
  200 ISTEP= 2                                                          INGR0480
  101 ACCEL = AMAG(RDD)                                                 INGR0490    7
      TTOGO =(TSTOPW-TW) +(TSTOPF-TF)                                   INGR0500
      TTOGO = TTOGO * CONV1                                             INGR0510
      IF(ISTART-1)1,1,2                                                 INGR0520
    2 DEL6  = ABS(COEF(8,1))                                            INGR0530
      DEL5 = ABS(COEF(7,I))                                             INGR0540
      DO 10 I=1,3                                                       INGR0550
      DEL5 = AMAX1(ABS(COEF(7,I)),DEL5)                                 INGR0560
   10 DEL6 = AMAX1(ABS(COEF(8,I)),DEL6)                                 INGR0570
      IF(DEL6 - .5*DEL5 ) 1,1,501                                       INGR0580
  501 DEL6 = DEL5                                                       INGR0590
    1 CALL HSIZE(TTOGO,DEL6,ACCEL,RXXMAG,ISTART,L,ISTEP,H )             INGR0600
C                                                                       INGR0610
C                                                                       INGR0620    22
      IF(ISTART-1) 20,20,30                                             INGR0630
   20 IF(LIMIT-8) 50,60,60                                              INGR0640
   60 LIMIT = 1                                                         INGR0650
   50 IF(IEND-1) 71,71,72                                               INGR0660
   71 HRK=H/2.                                                          INGR0670
      HH=HRK                                                            INGR0680
      CALL START                                                        INGR0690    31
      TIME=TIME+HRK                                                     INGR0700
      IF(NSTEPS.LE.14)  GO TO 40                                        INGR0705
   72 IF(ICHECK-1) 52,52,51                                             INGR0710
   51 ICHECK = 1                                                        INGR0720
```

```
        INGRT         -  EFN    SOURCE STATEMENT   -   IFN(S)   -

      IF(LIMIT-4) 162,161,162                                        INGR0730
  161 DO 163 I=1,3                                                   INGR0740
      RSTORE(I)=R(I)                                                 INGR0750
  163 VSTORE(I)=RD(I)                                                INGR0760
      GO TO 40                                                       INGR0770
  162 IF(LIMIT-7)  40,261, 40                                        INGR0780
  261 DO 263 I=1,3                                                   INGR0790
  263 RCHECK(I)=R(I)                                                 INGR0800
      IEND = 2                                                       INGR0810
      GO TO 40                                                       INGR0820
   52 ICHECK = 2                                                     INGR0830
      DO 22 I=1,3                                                    INGR0840
      STORE(LIMIT,I) = RDD(I)                                        INGR0850
   22 CONTINUE                                                       INGR0860
      LIMIT = LIMIT + 1                                              INGR0870
      IF(LIMIT-8) 40,25,25                                           INGR0880
   25 CALL DIFTAB(RSTORE,VSTORE,STORE,H,COEF )                       INGR0890     78
      IEND=1                                                         INGR0900
      DO 99 I=1,3                                                    INGR0910
   99 SAVE(I) = STORE(6,I)                                           INGR0920
      ISTART = 2                                                     INGR0930
      ICHECK = 1                                                     INGR0940
C                                                                    INGR0950
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * INGR0960
C                                                                    INGR0970
C     CONTINUE GAUSS INTEGRETION                                     INGR0980
   30 CALL INTEG(SAVE)                                               INGR0990     91
      HH=H                                                           INGR1000
      TIME = TIME + H                                                INGR1010
      CALL CONIC(RCO,VCO,TIME,XX,H,RCONIC,VCONIC)                    INGR1020     93
      DO 31 I=1,3                                                    INGR1030
      RXX(I) = R(I) - RCHECK(I)                                      INGR1040
      RCHECK(I) = R(I)                                               INGR1050
      RVEC(I) = RCONIC(I) + R(I)                                     INGR1060
   31 VVEC(I) = VCONIC(I) +RD(I)                                     INGR1070
      RXXMAG = AMAG(RXX)                                             INGR1080    112
   40 TF = TF +HH/86400.                                             INGR1090
```

```
      IF(TF-1.) 90,91,91                                          INGR1100
   91 TF = TF -1.                                                 INGR1110
      TW = TW + 1.                                                INGR1120
   90 CONTINUE
      RETURN                                                      INGR1135
      END                                                         INGR1140
```

SUBROUTINE INGRAT

COMMON VARIABLES

COMMON BLOCK // ORIGIN 00001 LENGTH 01014

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DATA	00000	R	DATA	00000	R	CON	00000	R
SAT	00017	R	SDA	00043	R	STT	00435	R
WRK	00606	R	CONV1	00012	R	PCØ	00641	R
VCØ	00644	R	RVEC	00661	R	VVEC	00664	R
R	00671	R	RD	00674	R	RDD	00677	R
TSTOPW	00702	R	TSTOPF	00703	R	H	00711	R
COEF	00714	R	TW	00667	R	TF	00670	R
TIME	00755	R	HRK	01000	R	A	01007	R
NSTEPS	01013	I						

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
RSTORE	01015	R	VSTORE	01020	R	STORE	01023	R
RCONIC	01050	R	VCONIC	01053	R	SAVE	01056	R
RCHECK	01061	R	RXX	01064	R			

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
ISTEP	01067	I	ACCEL	01070	R	TTOGO	01071	R
DELA	01072	R	DEL5	01073	R	I	01074	I
RXXMAX	01075	R	L	01076	I	LIMIT	01077	I
IEND	01100	I	HH	01101	R	ICHECK	01102	I
XX	01103	R						

ENTRY POINTS

INGPAT SECTION 4

SUBROUTINES CALLED

AMAG	SECTION	5	HSIZE	SECTION	6	START	SECTION	7
DIFTAB	SECTION	8	INTEG	SECTION	9	CONIC	SECTION	10
SYSLOC	SECTION	11						

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
100	3A	01127	200	5A	01132	101	6A	01134
1	21A	01233	2	9A	01157	10	15A	01206
501	20A	01231	20	25A	01252	30	90A	01410
50	28A	01263	60	27A	01256	71	30A	01264
72	36A	01304	40	113A	01456	52	65A	01340
51	38A	01310	162	53A	01325	161	41A	01315
163	47A	01320	261	55A	01330	263	59A	01331
22	72A	01351	25	77A	01363	99	84A	01377
31	106A	01444	90	116A	01475	91	115A	01467

DECK LENGTH IN OCTAL IS 00507.

2.3.1.3.1 Subroutine HSIZE

Purpose: HSIZE is a step-size, control routine designed to operate in conjunction with the Gauss-Jackson integration package. In its simplest form, HSIZE determines whether the stepsize (h) should be halved or whether it can be doubled without loss of accuracy or chance of producing a loop.

Deck Name: SIZE

Calling Sequence: HSIZE (TTOGO, DEL6, ACCEL, RXX, ISTART, L, IH, H)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	TTOGO	t_{go}	1	ARG	the time interval (sec) between the present epoch and the epoch of the next piece of observed data.
I	DEL6	Δ^6	1	ARG	the magnitude of the largest component of the vector composed of the sixth differences in the acceleration ($\Delta\ddot{\vec{r}}$ = Km/sec^2).
I	ACCEL	$\lvert\Delta\ddot{\vec{r}}\rvert$	1	ARG	the magnitude of the acceleration vector (Km/sec^2)
I	RXX	-	1	ARG	the magnitude of the change in the second integral compared to the previous step. (Km)
I	ISTART	-	1	ARG	an index used to define if INGRAT is functioning in the start mode (ISTART=1 for Runge-Kutta) or the continue

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
					mode (ISTART=2 for Gauss-Jackson)
I	L	-	1	ARG	an index used to check the number of consecutive times through HSIZE that the step size has been adjudged to be too small (when L=4 the step is doubled)
I	IH	-	1	ARG	an index used to determine if a step is to be guessed (to be refined later) or if the present step is to be checked. IH=1 for guess; =2 for present step
O	H	h	1	ARG	the step size to be employed in the numerical integration of the equations of motion employing the Gauss-Jackson backwards difference formulation. (sec)

Subroutines Required: None

Functions Required: ALOG10 (log to the base 10)

Approximate Deck Length: 377 (octal)

Discussion:

The basic rationale for deciding whether the stepsize for the integration (Gauss) is adequate is based on a set of more or less arbitrary limits for the absolute value of the largest sixth difference (DEL6). These limits, in turn, are based on the order of the magnitude of the acceleration and were selected in such a manner that the integration series would be convergent to the desired degree with terms through the sixth difference included.

One slight problem arises in the application of this rationale in that if the acceleration itself ever passes through zero, the tolerance goes to zero; and the stepsize is immediately adjudged to be too large. To surmount this

problem, a second test is employed in those cases where the quantity DEL6 is larger than that allowed by the upper limit. This second test requires that the contribution of the sixth difference to the second integral be less than one part in 10^4 of the change in the second integral for the last step, i.e.

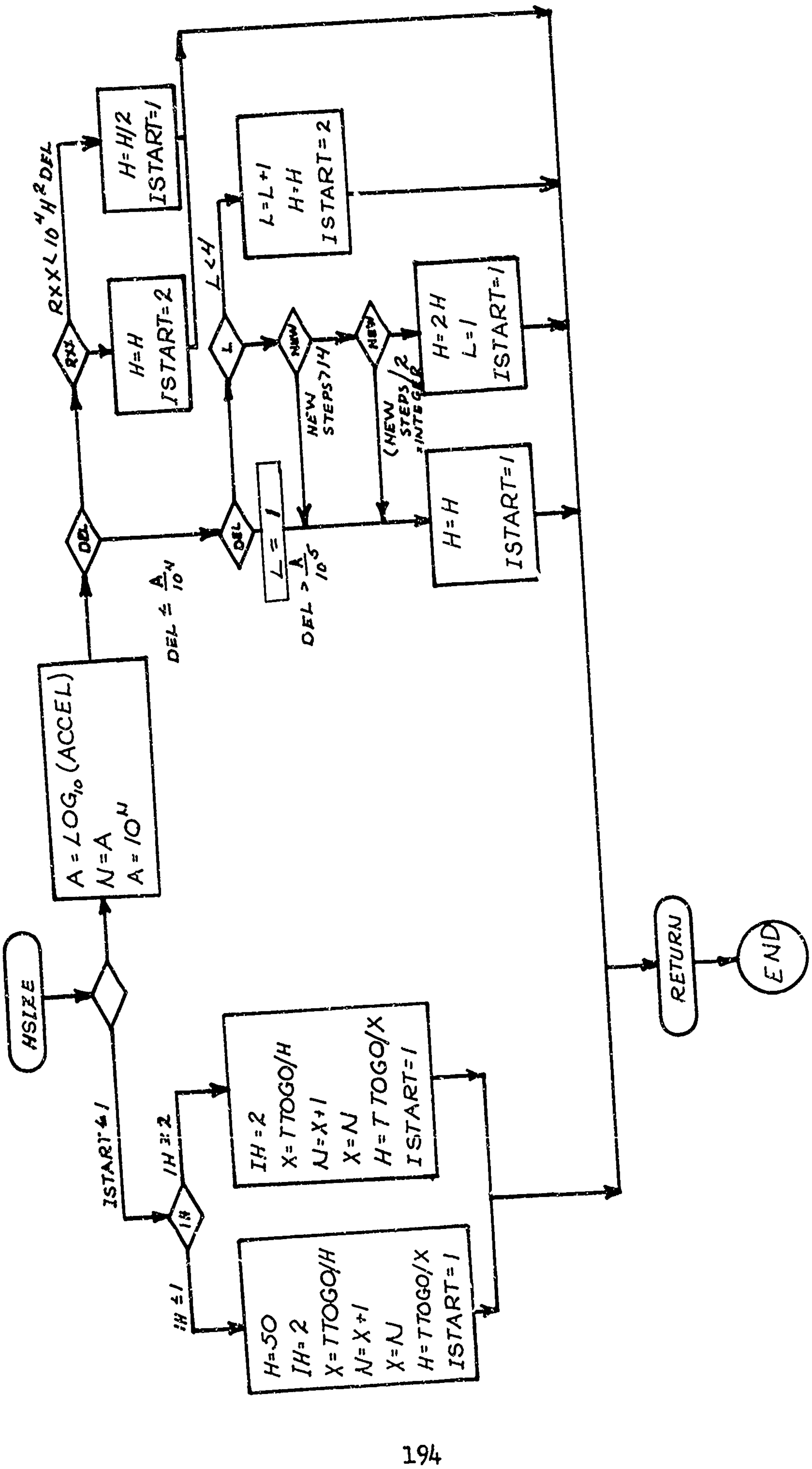
$$\Delta r^{6th} \approx H^2 \Delta^6 / 15$$
$$\lesssim 10^{-4} \left| \Delta \vec{r}_{n-1} - \Delta \vec{r}_{n-2} \right|$$

or say

$$10^3 H^2 \Delta^6 \lesssim \left| \Delta \vec{r}_{n-1} - \Delta \vec{r}_{n-2} \right|$$

(The second integral is the displacement from the reference conic which is growing in a reasonably slow fashion.) The step is always halved if the upper tolerance level is violated.

Three other modifications to this basic rationale were also coded for the sake of operation, efficiency and accuracy. Each of the three pertains to the case where a signal would normally be generated to double the step size (DEL6 less than lower limit). The first modification is employed to assure that the step size is too small at several consecutive points before a signal is generated to double the step. This practice assures that fewer cases in which the tolerance range is temporarily violated on the low end will result in restart. The second modification is employed to assure that after a whole number of the new integration steps, the time-to-go can be reduced to zero (this performance is essential in the operation of the program). This second test is passed automatically if the step is halved; however, if the step is doubled, time-to-go divided by the old step must be even. The third modification is employed to assure that the integration can be effectively restarted prior to the time that the time-to-go will reach zero.


HSIZE
A = LOG10 (ACCEL)
N=A
A = 10^N
DEL ≤ A/10^N
DEL
RXX
RXX L 10^4 H^2 DEL
H=H
ISTART=2
H=H/2
ISTART=1
L<4
L
L=L+1
H=H
ISTART=2
DEL
L = 1
DEL > A/10^5
NEW
NEW STEPS>14
NEW
(NEW STEPS/2 = INTEGER
H=2H
L=1
ISTART=1
H=H
ISTART=1
ISTART≠1
IH
IH=2
IH=2
X=TTOGO/H
N=X+1
X=N
H=TTOGO/X
ISTART=1
IH≠1
H=50
IH=2
X=TTOGO/H
N=X+1
X=N
H=TTOGO/X
ISTART=1
RETURN
END

```
      SUBROUTINE HSIZE( TTOGO,DEL6,ACCEL,RXX,ISTART,L,IH,H )           SIZE0020
C                                                                       SIZE0030
C     HSIZE DETERMINES THE STEP SIZE TO BE UTILIZED IN NUMERICAL       SIZE0040
C              INTEGRATION VIA GAUSS METHOD. THE SELECTED INTERVAL      SIZE0050
C              WILL BE DOUBLED OR HALVED IF SUCH IS CONSISTENT WITH THE SIZE0060
C              PRINT TIME (OR TERMINAL TIME ) CONSTRAINT, IF THE SIXTH  SIZE0070
C              DIFFERENCE FALLS OUTSIDE A REGION DETERMINED BY THE ORDERSIZE0080
C              OF THE ACCELERATION ,AND IF THE CONTRIBUTION OF THE      SIZE0090
C              SIXTH DIFFERENCE TO THE SECOND INTEGRAL IS LESS THAN A   SIZE0100
C              SMALL FRACTION OF THE CHANGE IN RXX FROM THE PREVIOUS H. SIZE0110
C     TTOGO = TIME REMAINING FOR INTEGRATION                            SIZE0120
C     DEL6  = SIXTH DIFFERENCE OF ACCELERATION                          SIZE0130
C     ACCEL = MAGNITUDE OF ACCELERATION VECTOR                          SIZE0140
C     RXX   = CHANGE IN THE SECOND INTEGRAL FOR PREVIOUS STEP           SIZE0150
C     ISTART= 1 IF INTEG IS TO COMMAND A RESTART ** =2 IF INTEG         SIZE0160
C              IS TO CONTINUE                                           SIZE0170
C     L     = COUNT TO SEE HOW MANY TIMES THAT DEL5 IS LESS THAN LIMIT  SIZE0180
C              WHEN L=1...3 STEP SIZE REMAINES CONSTANT                 SIZE0190
C              WHEN L= 4 H IS DOUBLED THEN L IS RESET TO 1              SIZE0200
C     IH    = 1 FOR INITIAL GUESS OF H                                  SIZE0210
C           = 2 FOR NO COMPUTATION OF H                                 SIZE0220
C           = 3 TO BASE STEP ON PRESENT VALUE OF H                      SIZE0230
C     H     = STEP SIZE FOR GAUSS INTEGRATION ( H RUNGE-KUTTA = H/2)    SIZE0240
C                                                                       SIZE0250
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * SIZE0260
C                                                                       SIZE0270
      IF(ISTART-1) 50,50,2                                              SIZE0280
   50 IF(IH-2) 1,30,30                                                  SIZE0290
    1 H=50.                                                             SIZE0300
      X= TTOGO/H                                                        SIZE0310
      N= X+1.                                                           SIZE0320
      X = N                                                             SIZE0330
      H = TTOGO/X                                                       SIZE0340
      GO TO 10                                                          SIZE0350
    2 SCALE = ALOG10(ABS(ACCEL))                                        SIZE0360    9
      N = SCALE                                                         SIZE0370
```

```
     SIZE        -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      SCALE1 = 10.**N*.2E-3                                           SIZE0380    11
      SCALE2 = SCALE1/40.                                             SIZE0390
      ABSDEL = ABS(DEL6)                                              SIZE0400
      CHECK6 = 1.E3 *H*H*ABSDEL                                       SIZE0410
      IF(ABSDEL - SCALE1) 5,5,6                                       SIZE0420
    6 IF/RXX - CHECK6) 22,22,5                                        SIZE0430
   22 H = H/2.                                                        SIZE0440
      GO TO 10                                                        SIZE0450
    5 IF( ABSDEL - SCALE2 ) 7,7,8                                     SIZE0460
    8 H=H                                                             SIZE0470
      L = 1                                                           SIZE0480
      GO TO 20                                                        SIZE0490
    7 IF(L-4 ) 11,12,12                                               SIZE0500
   11 L=L+1                                                           SIZE0510
      H=H                                                             SIZE0520
      GO TO 20                                                        SIZE0530
   12 RATIO = TTOGO/(2.*H)                                            SIZE0540
      Y= RATIO + .1                                                   SIZE0550
      X= Y + .6                                                       SIZE0560
      IY= Y                                                           SIZE0570
      IX = X                                                          SIZE0580
      IF(IX-14) 20,20,80                                              SIZE0590
   80 IF( IY-IX) 13,14,14                                             SIZE0600
   14 H =2.*H                                                         SIZE0610
      L = 1                                                           SIZE0620
      GO TO 10                                                        SIZE0630
   13 H = H                                                           SIZE0640
      GO TO 20                                                        SIZE0650
   10 ISTART = 1                                                      SIZE0660
   30 CONTINUE                                                        SIZE0670
      RETURN                                                          SIZE0700
   20 ISTART = 2                                                      SIZE0710
      GO TO 30                                                        SIZE0720
      END                                                             SIZE0730
```

SUBROUTINE HSIZE

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
X	00001	R	N	00002	I	SCALE	00003	R
SCALE1	00004	R	SCALE2	00005	R	ABSDEL	00006	R
CHECK6	00007	R	RATIO	00010	R	Y	00011	R
IY	00012	I	IX	00013	I			

ENTRY POINTS

HSIZE SECTION 3

SUBROUTINES CALLED

ALOG10	SECTION	4	.XP2.	SECTION	5	E.1	SECTION	6
E.2	SECTION	7	E.3	SECTION	8	E.4	SECTION	9
CC.1	SECTION	10	CC.2	SECTION	11	CC.3	SECTION	12
CC.4	SECTION	13	SYSLOC	SECTION	14			

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
50	3A	00041	2	8A	00074	1	5A	00045
30	37A	00246	10	36A	00244	5	17A	00152
6	13A	00142	22	15A	00146	7	21A	00163
8	19A	00156	20	38A	00247	11	23A	00167
12	26A	00175	80	30A	00227	13	34A	00241
14	32A	00233						

DECK LENGTH IN OCTAL IS 00335.

2.3.1.3.2 Subroutine START

Purpose: START is a fourth order Runge-Kutta integration routine utilized to establish the accelerations at the seven points along the trajectory and define the difference table required by the Gauss-Jackson integration.

Deck Name: START

Calling Sequence: Call START

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	RCO VCO	$\vec{r}_{c_o}$ $\vec{v}_{c_o}$	3 3	WRK (28) WRK (31)	The position and velocity vectors in the frame of 1950.0 describing the conic reference trajectory at the time of the last rectification (Km, Km/sec)
O	RAD VEL	$\vec{r}_{50}$ $\vec{v}_{50}$	3 3	WRK (44) WRK (47)	the position and velocity vectors in the computational coordinate frame of 1950.0 (Km, Km/sec) at the epoch t+h
I/O	RDD RD R	$\Delta\ddot{\vec{r}}$ $\Delta\dot{\vec{r}}$ $\Delta\vec{r}$	3 3 3	WRK (58) WRK (55) WRK (52)	the acceleration (Km/sec^2) the velocity (Km/sec), and the position (Km) vectors relative to the reference ellipse in the coordinate frame of 1950.0
I/O	TIME	$t-t_c$	1	WRK (104)	the time interval in seconds since the last rectification of the reference ellipse

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
O	RC	$\vec{r}_c$	3	WRK (117)	the instantaneous reference position
	VC	$\vec{v}_c$	3	WRK (120)	and velocity vectors (1950.0) (Km,Km/sec)
I	H	h	1	WRK (123)	the step size employed in the integration (sec). This quantity is one-half that employed in the Gaussian integration.

Subroutines Required: CONIC (reference conic)
MOTION (total acceleration vector relative to elliptical reference)

Functions Required: None

Approximate Deck Length: 500 (octal)

Discussion:

START functions in the standard single integration Runge-Kutta mode. That is, the second-order differential equation is reduced to two first-order differential equations with the substitution

$$\dot{\vec{v}} = \ddot{\vec{r}}$$
$$\dot{\vec{r}} = \vec{v}$$

and each part which may be represented as

$$\dot{x} = f\ (t,x)$$

as integrated as follows:

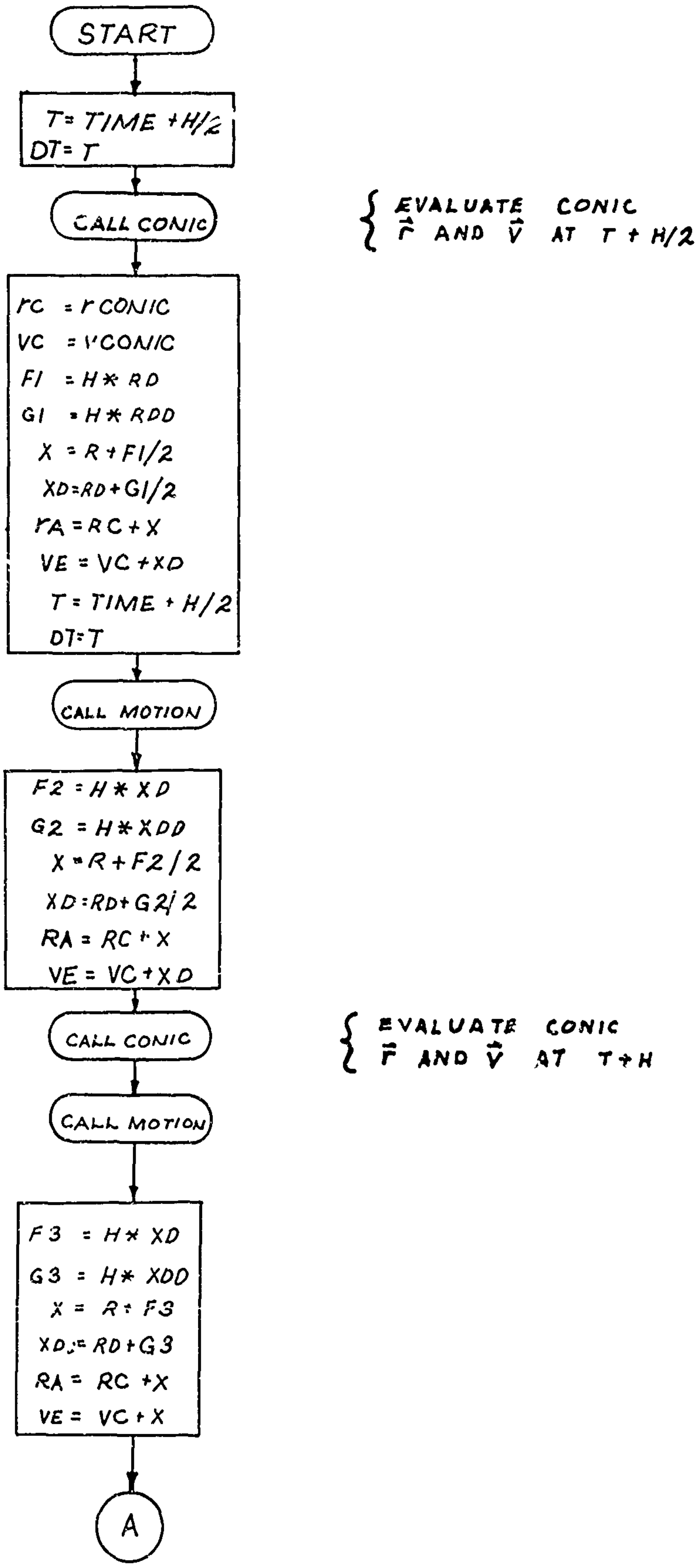
$$x\ (t+h) = x\ (t) + \frac{h}{6}\ (f_1 + 2f_2 + 2f_3 + f_4)$$

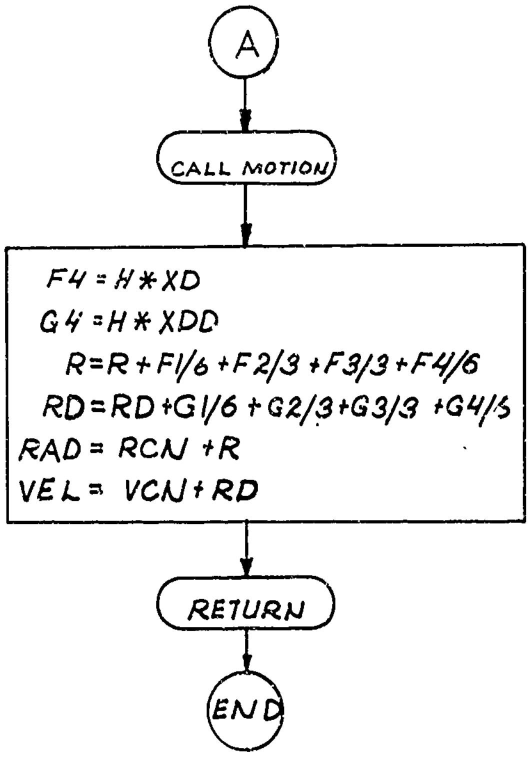
where:

$$f_1 = f\ (t,x)$$
$$f_2 = f\ (t+\frac{h}{2},\ x+\frac{h}{2}\ f_1)$$
$$f_3 = f\ (t+\frac{h}{2},\ x+\frac{h}{2}\ f_2)$$
$$f_4 = f\ (t+h,\ x+hf_3)$$

No corrector cycles or smoothing operations are superimposed on this procedure. Thus, since the errors tend to grow more rapidly in this method of integration than is the case with the Gauss-Jackson integration, it was decided that the step size utilized for each step of the starter would be no more than one-half of that utilized for the continuation integration. This step (made in INGRAT) assures a more consistent level of accuracy in the two routines. It is noted, however, that the subroutine MOTION must now be called eight times in this mode of operation, compared to one time in the Gauss mode.

A complete discussion of the development of the Runge-Kutta family of integration formulae can be found in many texts treating the subject of numerical integration. One such text is Numerical Analysis by K. Z. Kunz and published by McGraw-Hill in 1957.


START
T= TIME +H/2
DT= T
CALL CONIC
EVALUATE CONIC
r AND V AT T + H/2
rC = r CONIC
VC = VCONIC
F1 = H* RD
G1 = H* RDD
X = R + F1/2
XD=RD+G1/2
rA = RC + X
VE = VC +XD
T = TIME + H/2
DT=T
CALL MOTION
F2 = H* XD
G2 = H*XDD
X = R+F2/2
XD =RD+ G2/2
RA = RC + X
VE = VC + XD
CALL CONIC
EVALUATE CONIC
r AND V AT T+H
CALL MOTION
F3 = H* XD
G3 = H* XDD
X = R+ F3
XD = RD+G3
RA = RC +X
VE = VC + X
A


A
CALL MOTION
F4 = H*XD
G4 = H*XDD
R=R+F1/6+F2/3+F3/3+F4/6
RD=RD+G1/6+G2/3+G3/3+G4/6
RAD = RCN +R
VEL = VCN+RD
RETURN
END

```
      SUBROUTINE START                                                  START020
C                                                                       START030
C     START SETS-UP A 4-TH ORDER RUNGE-KUTTA INTEGRATION AND EVALUATES  START040
C             THE POSITIONS AND VELOCITIES NECESSARY TO COMPUTE THE     START050
C             ACCELERATIONS USED IN A GAUSS-JACKSON INTEGRATION .       START060
C     THE VERSION USED IN THIS ROUTINE  IS THE KUTTA-SIMPSON ONE-THIRD  START070
C             RULE EXTENSION OF THE QUADRATURE FORMULA                  START080
C     RAD = RADIUS VECTOR (1950.0)                                      START090
C     VEL = VELOCITY VECTOR(1950.0)                                     START100
C     RDD = ACCELERATION VECTOR FOR RELATIVE MOTION (1950.0)            START110
C     RD  = RELATIVE VELOCITY VECTOR (1950.0 )                          START120
C     R= RELATIVE POSITION VECTOR (1950.0)                              START130
C     TIME= SECONDS FROM LAST RECTIFICATION                             START140
C     H   = STEP SIZE IN SECONDS                                        START150
C                                                                       START160
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * START170
C                                                                       START180
      DIMENSION  CON(1), SAT(1), SDA(1), WRK(1), STT(1)                 START190
C                                                                       START200
      COMMON DATA                                                       START210
C                                                                       START220
      EQUIVALENCE  (DATA(  1),CON), (DATA( 16),SAT), (DATA( 36),SDA)    START230
     2              ,(DATA(286),STT), (DATA(391),WRK)                   START240
C                                                                       START250
C                                                                       START260
      DIMENSION  RAD(3)     ,VEL(3)    ,R(3)      ,RD(3)     ,RDD(3)    START270
     2,RC(3)     ,VC(3)     ,RA(3)     ,VE(3)     ,X(3)      ,XD(3)     START280
     3,XDD(3)    ,RR(3)     ,RDK(3)    ,F1(3)     ,F2(3)     ,F3(3)     START290
     4,F4(3)     ,G1(3)     ,G2(3)     ,G3(3)     ,G4(3)     ,A(4)      START300
     5,RDDR(3)                                                          START305
C                                                                       START310
      EQUIVALENCE                                                       START320
     2   (WRK( 28),RCO   ), (WRK( 31),VCO   ), (WRK( 44),RAD   )        START330
     3  ,(WRK( 47),VEL   ), (WRK( 52),R     ), (WRK( 55),RD    )        START340
     4  ,(WRK( 58),RDD   ), (WRK(123),H     ), (WRK(120),VC    )        START350
     5  ,(WRK(130),A     ), (WRK(104),TIME  ), (WRK(117),RC    )        START360
```

```
     ****                                                                     12/03/85              PAGE  12
         STRT        -  EFN   SOURCE STATEMENT  -  IFN(S)  -

C                                                                             START370
      EQUIVALENCE           (      RAD,RA     ), (     VEL,VE     )           START380
     2  ,(        R,X      ), (      RD,XD     ), (     RDD,XDD    )           START390
C                                                                             START400
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *         START410
C     .                                                                       START420
      DO 10 I=1,3                                                             START430
      RR(I) = R(I)                                                            START440
      RDDR(I) = RDD(I)                                                        START445
   10 RDR(I) = RD(I)                                                          START450
      T     = TIME + H/2.                                                     START460
      DT = T                                                                  START470
      CALL CONIC(RCO,VCO,T,O,DT,RC,VC)                                        START480      14
      DO 1 I=1,3                                                              START490
      F1(I) = H* RD(I)                                                        START500
      G1(I) = H*RDD(I)                                                        START510
      X (I) =RR(I) + F1(I)/2.                                                 START520
      XD(I)=RDR(I) + G1(I)/2.                                                 START530
      RA (I)= RC (I) + X(I)                                                   START540
    1 VE (I)= VC (I) +XD(I)                                                   START550
      CALL MOTION(1,2)                                                        START560
C                                                                             START570
C                                                                             START580      38
      DO 2 I=1,3                                                              START590
      F2(I) = H* XD(I)                                                        START600
      G2(I) = H*XDD(I)                                                        START610
      X (I) =RR(I) + F2(I)/2.                                                 START620
      XD(I)=RDR(I) + G2(I)/2.                                                 START630
      RA (I)= RC (I) + X(I)                                                   START640
    2 VE (I)= VC (I) +XD(I)                                                   START650
      CALL MOTION(1,2)                                                        START660
C                                                                             START670
C                                                                             START680      62
      T = TIME + H                                                            START690
      DT     = T                                                              START700
      CALL CONIC(RCO,VCO,T,O,DT,RC,VC)                                        START710      64
      DO 3 I=1,3                                                              START720
```

```
      STRT         -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      F3(I) = H* XD(I)                                          START730
      G3(I) = H*XDD(I)                                          START740
      X (I) =RR(I) + F3(I)                                      START750
      XD(I)=RDR(I) + G3(I)                                      START760
      RA (I)= RC (I) + X(I)                                     START770
    3 VE (I)= VC (I) +XD(I)                                     START780
      CALL MOTION(1,2)                                          START790
C                                                               START800
C                                                               START810    88
      DO 4 I=1,3                                                START820
      F4(I) = H* XD(I)                                          START830
      G4(I) = H*XDD(I)                                          START840
      R(I)  =RR(I) + (F1(I)+2.*(F2(I)+F3(I))+F4(I))/6.          START850
      RD(I) =RDR(I)+ (G1(I)+2.*(G2(I)+G3(I))+G4(I))/6.          START860
      RAD(I)= RC (I) + R(I)                                     START870
      RDD(I) = RDDR(I)                                          START875
    4 VEL(I)= VC (I) +RD(I)                                     START880
C                                                               START890
C                                                               START900
      RETURN                                                    START910
      END                                                       START920
```

****
STRT

12/03/85 PAGE 22

STORAGE MAP

SUBROUTINE START

COMMON VARIABLES

COMMON BLOCK // ORIGIN 00001 LENGTH 01013

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DATA	00000	R	DATA	00000	R	CON	00000	R
SAT	00017	R	SDA	00043	R	STT	00435	R
WRK	00606	R	RCD	00641	R	VCO	00644	R
RAD	00661	R	VEL	00664	R	R	00671	R
RD	00674	R	RDD	00677	R	H	01000	R
VC	00775	R	A	01007	R	TIME	00755	R
RC	00772	R	RA	00661	R	VE	00664	R
X	00671	R	XD	00674	R	XDD	00677	R

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
RR	01014	R	RDR	01017	R	F1	01022	R
F2	01025	R	F3	01030	R	F4	01033	R
G1	01036	R	G2	01041	R	G3	01044	R
G4	01047	R	RDDR	01052	R			

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
T	01055	R	DT	01056	R	U	01057	R

ENTRY POINTS

START SECTION 4

SUBROUTINES CALLED

CONIC SECTION 5   MOTION SECTION 6   SYSLOC SECTION 7

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
10	9A	01074	1	32A	01145	2	56A	01203
3	82A	01254	4	114A	01326			

DECK LENGTH IN OCTAL IS 00337.

2.3.1.3.3 Subroutine DIFTAB

Purpose: DIFTAB differences seven values of the acceleration vector and constructs the diagonal of backward differences and leading sums required in the Gauss-Jackson integration routine (INTEG).

Deck Name: DIFTAB

Calling Sequence: CALL DIFTAB (RSTORE, VSTORE, STORE, H, COEF)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	RSTORE VSTORE	$\Delta\vec{r}_4$ $\Delta\vec{v}_4$	3 3	Arg Arg	the position and velocity vectors relative to the reference ellipse at the center point in the table. (Km, Km/sec)
I	STORE	$\Delta\ddot{\vec{r}}_1$	7, 3	Arg	the array of acceleration vectors at seven times (spaced at intervals of h) which are to be differenced and summed (Km/sec^2)
I	H	h	1	Arg	the step in time which correlates the various entries in the STORE array (sec)
O	COEF	Σ_i^2 Σ_i^1 Δ_i^1 $\vdots$ Δ_i^6	8, 3	Arg	the array of numbers which is required by INTEG for the integration of the equations of Motion. The first subscript denotes the second sum, the first sum, the first difference ..., the sixth difference. The second subscript denotes the x, y and z components.

Subroutines Required: None

Functions Required: None

Approximate Deck Length: 470 (octal)

Discussion:

The output of the start cycle is an array of acceleration vectors at seven times (spaced h seconds apart) and position and velocity vectors for the center point. These numbers will be utilized to construct the leading diagonal required in the Gauss-Jackson integration process as follows

1) The six differences of the components of the acceleration vector will be constructed.

2) The second sum corresponding to the position for the central value of the acceleration is calculated from:

$$\Sigma_0^2 = \frac{X_0}{h^2} - \frac{\ddot{X}_0}{12} + \frac{\Delta_0^2}{240} - \frac{31\Delta_0^4}{60480}$$

3) The average value of the first sum corresponding to the velocity for the central value of the acceleration is calculated from:

$$\Sigma_0' = \frac{X_0}{h} + \frac{(\Delta^1_{-1/2} + \Delta^1_{+1/2})}{24} - \frac{11(\Delta^3_{-1/2} + \Delta^3_{+1/2})}{1440} + \frac{190(\Delta^5_{-1/2} + \Delta^5_{+1/2})}{120960}$$

and the corresponding first sum on the proper diagonal in the difference table is

$$\Sigma_{1/2}' = \Sigma_0' + \tfrac{1}{2}\ddot{X}_0$$

4) The first and second sums are stepped down to the next to last diagonal and the terms along this diagonal are loaded into an array set aside for input into INTEG. (The next to last diagonal was selected since INTEG updates the diagonal before performing the integration).

In order to preserve accuracy in these operations the differencing will be performed in double precision. This procedure has been shown to conserve approximately one significant figure when operating with the IBM 7094.

Computational Logic:

DIFTAB

CONSTRUCT DIFFERENCE TABLE FOR EACH COMPONENT OF $\ddot{\vec{r}}$

USING THE POSITION AND VELOCITY VECTORS AT THE CENTER POINT IN THE TABLE, EVALUATE Σ', Σ^2 AND STEP THESE SUMS DOWN TO THE DIAGONAL OF THE NEXT TO LAST ACCELERATION. THE GAUSS CONTINUATION WILL STEP ALL NUMBERS TO THE NEXT DIAGONAL BEFORE STEPPING THE FUNCTION

EQUATE THE COEFFICIENT ARRAY (DIMENSIONED 8 BY 3) TO THE DIAGONAL CONTAINING THE NEXT TO LAST VALUE OF $\ddot{X}$, $\ddot{Y}$, $\ddot{Z}$ (THIS VECTOR CORRESPONDS TO $\vec{R}$, $\vec{V}$ AT THE LAST STEP TAKEN BY THE STARTER ROUTINE)

RETURN

END

```
      SUBROUTINE DIFTAB(RSTORE,VSTORE,STORE,H,COEF)                      DTAB0020
C                                                                        DTAB0030
C     THIS ROUTINE IS USED IN STARTING AN INTEGRATION . ITS PURPOSE IS   DTAB0040
C            TO ESTIMATE THE TERMS INTHE DIFFERENCE TABLE AND COMPUTE    DTAB0050
C            THE FIRST AND SECOND SUMS .                                 DTAB0060
C     RSTOR,VSTOR= INITIAL CONDITIONS FOR R AND RDOT                     DTAB0070
C     STORE = ARRAY OF ACCELERATION VECTORS ( 7 POINTS )                 DTAB0080
C     COEF = ARRAY OF SUMS AND DIFFERENCES **1=I SUM,2=1 SUM ...ETC      DTAB0090
C                                                                        DTAB0100
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    DTAB0110
      DIMENSION COEF(8,3),RSTORE(3),VSTORE(3),STORE(7,3)                 DTAB0120
C                                                                        DTAB0130
      DOUBLE PRECISION A(7),B(6),C(5),D(4),E(3),F(2)                     DTAB0140
C                                                                        DTAB0150
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    DTAB0160
C                                                                        DTAB0170
      DO 200 K=1,3                                                       DTAB0180
      POSIT = RSTORE(K)                                                  DTAB0190
      VELOC = VSTORE(K)                                                  DTAB0200
      DO 100 I=1,7                                                       DTAB0210
  100 A(I) = STORE(I,K)                                                  DTAB0220
      DO 1 I=1,6                                                         DTAB0230
    1 B(I)=A(I+1)-A(I)                                                   DTAB0240
      DO 2 I=1,5                                                         DTAB0250
    2 C(I)=B(I+1)-B(I)                                                   DTAB0260
      DO 3 I=1,4                                                         DTAB0270
    3 D(I)=C(I+1)-C(I)                                                   DTAB0280
      DO 4 I=1,3                                                         DTAB0290
    4 E(I)=D(I+1)-D(I)                                                   DTAB0300
      DO 5 I=1,2                                                         DTAB0310
    5 F(I)=E(I+1)-E(I)                                                   DTAB0320
C                                                                        DTAB0330
C                                                                        DTAB0340
      SUM2 =POSIT/(H*H) -A(4)/12. + C(3)/240. - 31.* E(2)/60480.         DTAB0350
     1         + 289.*(F(2)-F(1))/3628800.                               DTAB0360
      SUM1 =VELOC/ H  +(B(3)+B(4))/24. - 11.*(D(2)+D(3))/1440.           DTAB0370
```

```
 1        + 191.*(F(1)+F(2))/120960.                                   DTAB0380
      SUM1  = SUM1 +.5* A(4)                                           DTAB0390
      SUM2  = SUM2 + SUM1                                              DTAB0400
      DO 300 I=1,2                                                     DTAB0410
      SUM1 = SUM1 + A(I+4)                                             DTAB0420
      SUM2 = SUM1 + SUM2                                               DTAB0430
  300 CONTINUE                                                         DTAB0440
      COEF(8,K) = F(2)-F(1)                                            DTAB0450
      COEF(7,K) = F(1)                                                 DTAB0460
      COEF(6,K) = E(2)                                                 DTAB0470
      COEF(5,K) = D(3)                                                 DTAB0480
      COEF(4,K) = C(4)                                                 DTAB0490
      COEF(3,K) = B(5)                                                 DTAB0500
      COEF(2,K) = SUM1                                                 DTAB0510
      COEF(1,K) = SUM2                                                 DTAB0520
  200 CONTINUE                                                         DTAB0530
      RETURN                                                           DTAB0540
      END                                                              DTAB0550
```

SUBROUTINE DIFTAB

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
A	00001	D	B	00017	D	C	00033	D
D	00045	D	E	00055	D	F	00063	D

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
K	00067	I	POSIT	00070	R	VELOC	00071	R
SUM2	00072	R	SUM1	00073	R			

ENTRY POINTS

DIFTAB SECTION 3

SUBROUTINES CALLED

SYSLOC SECTION 4

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
200	61A	00375	100	9A	00164	1	15A	00174
2	22A	00204	3	29A	00214	4	36A	00224
5	43A	00234	300	51A	00351			

DECK LENGTH IN OCTAL IS 00470.

2.3.1.3.4 Subroutine INTEG

Purpose: INTEG continues a stepwise integration of the equations of motion in a Gauss-Jackson mode once started by an independent process. The routine employes six diffences and is formulated around the backwards difference concept (no corrections)

Deck Name INTG

Calling Sequence: CALL INTEG (SAVE)

Input/Output

I/O	FORTRAN NAME	MATH NAME	DIMENSION	COM/ ARG	DEFINITION
I/O	COEF	$\Sigma^2 \ldots \Delta^6$	8,3	WRK(71)	The array containing the diagonal to be utilized in the next integration step employing Gauss' Equation (2 sums, 6 differences, 3 components of acceleration)
I	RDD	$\Delta\ddot{\vec{r}}$	3	WRK(58)	The acceleration vector defining the motion relative to the reference ellipse (Km/sec^2)
I/O	SAVE	$\Delta\ddot{\vec{r}}_{-1}$	3	Arg	The acceleration vector at the last integration step (Km/sec^2)
O	RD R	$\Delta\dot{\vec{r}}$ $\Delta\vec{r}$	3	WRK(52) WRK(58)	The incremented position and velocity vectors for the motion relative to the reference ellipse (Km/sec^2)

I/O	FØRTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	STEP	H	1	WRK(68)	The stepsize for the numerical integration of the eq. of motion (sec)

Subroutines Required: None

Functions Required: None

Approximate Deck Length: 330 (octal)

Discussion

The continuation of the intregration will be performed by employing a Gauss-Jackson or double sum formulation. This process is accomplished by mechanizing the following backwards difference formulae

$$\dot{x}_{i+1} = H\Big(\Sigma'_{i+1/2} + \frac{\ddot{x}}{2}i + \frac{5}{12}\Delta'_{i-1/2} + \frac{3}{8}\Delta^2_{i-1}$$

$$+ \frac{251}{720}\Delta^3_{i-3/2} + \frac{95}{288}\Delta^4_{i-2} + \frac{19087}{60480}\Delta^5_{i-5/2}$$

$$+ \frac{5257}{17280}\Delta^6_{i-3}\Big)$$

$$x_{i+1} = H^2\Big(\Sigma^2_{i+1} + \frac{\ddot{x}}{12}i + \frac{\Delta'}{12}i-1/2 + \frac{19}{240}\Delta^2_{i-1}$$

$$+ \frac{18}{240}\Delta^3_{i-3/2} + \frac{1726}{24192}\Delta^4_{i-2} + \frac{1650}{24192}\Delta^5_{i-5/2}$$

$$+ \frac{15}{240}\Delta^6_{i-3}\Big)$$

$$\Sigma'_{i+1/2} = \Sigma'_{i-1/2} + \ddot{x}$$

$$\Sigma^2_{i+1} = \Sigma^2_i + \Sigma'_{i+1/2}$$

$$\Delta'_{i-1/2} = \ddot{x}_i - \ddot{x}_{i-1}$$

$$\Delta^2_{i-1} = \Delta'_{i-1/2} - \Delta'_{i-3/2}$$

This mechanization will be aided by the establishment of a two dimensional array which will be referred to as the coefficient array (COEF) and which will contain the information required by the formula.

$$COEF(1,1) = \Sigma_x^2 \qquad COEF(1,2) = \Sigma_y^2 \ldots$$

$$COEF(2,1) = \Sigma_x'$$

$$COEF(3,1) = \Delta_x'$$

$$\vdots$$

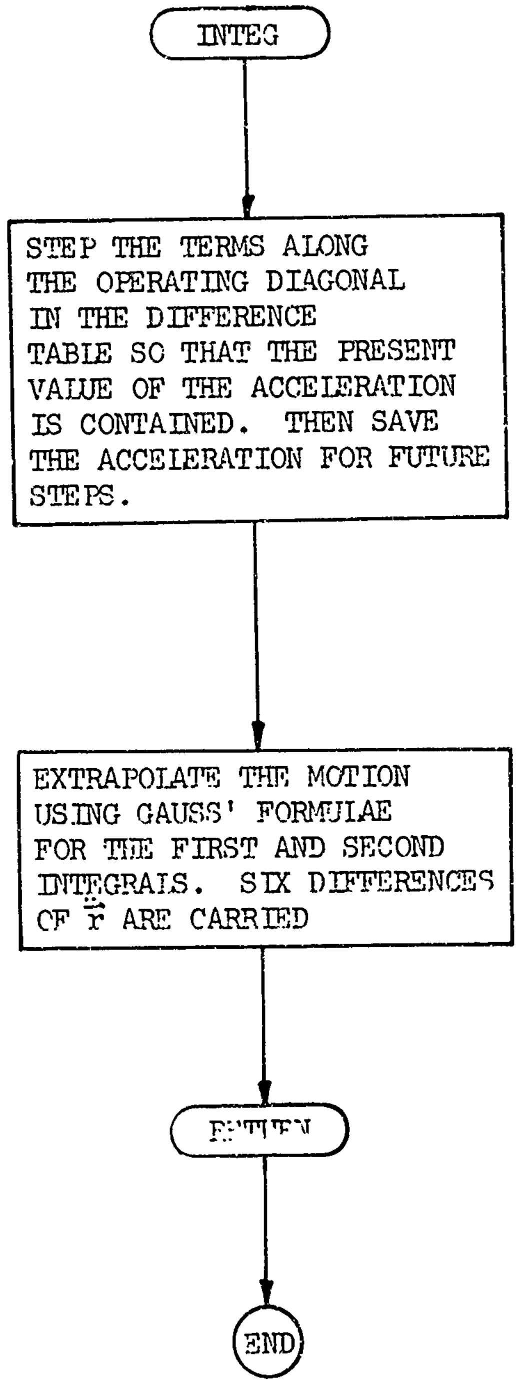
$$COEF(8,1) = \Delta_x'$$

(This array is thus seen to contain the terms along the second leading diagonal) Upon entry to INTEG, the diagonal which is stored in COEF will correspond to the last integration step; thus, the first task to be performed will be the updating of COEF according to the definition of its members. This done, the trajectory can be stepped.

Two specific comments are required regarding this process. First, the process is obviously not exact and thus for extreme accuracy a corrector cycle based on central differcences should be employed. This type of operation has been avoided by attention to detail in the formulation of the trajectory portion of the program and by careful selection of the stepsize. However, this revision is felt to be advisable if any numerical problems develop during the application of the differential corrections program to the determination of satellite orbits.

The other comment is an observation which pertaines to the fact that two integrations which are performed are both based on the acceleration rather than on the acceleration and velocity. This fact results in the evaluation of the second integral to an accuracy superior to that obtained with the other approach or for that matter, to that obtained for the first integral. This behavior is very desirable for this program since it assures that the major contributives to the acceleration at future points along the trajectory (primarily functions of position) will be well known. Care must be taken, however, in applying this logic to trajeccories for which the non-linear effects of drag, etc. are involved since for these cases the accumulative errors in the first integral ($\dot{X}$) could drastically effect the predicted path.

Computational Logic:



```
      SUBROUTINE INTEG(SAV)                                                GAUSS020
C                                                                          GAUSS030
C     THIS ROUTINE PERFORMS A CONTINUED INTEGRATION OF A SECOND ORDER      GAUSS040
C             DIFFERENTIAL EQUATION . OUTPUT INCLUDES FIRST AND            GAUSS050
C             SECOND INTEGRALS VIA A GAUSS JACKSON-SECOND SUM PROCESS.     GAUSS060
C     COEF ARRAY    CONTAINES THE SUMS AND DIFFERENCES REQUIRED TO KEEP    GAUSS070
C             THE INTEGRATION GOING.(THESE COEFFICIENTS ARE STEPPED        GAUSS080
C             BEFORE RETURNING TO THE MAIN PROGRAM).                       GAUSS090
C     THE QUANTITIES RDD,RD AND R REPRESENT THE ACCELERATION,VELOCITY      GAUSS100
C             AND POSITION FOR THE FUNCTION BEING INTEGRATED . RDD IS      GAUSS110
C             EVALUATED AT PRESENT POINT , RD AND R AT NEXT POINT .        GAUSS120
C                                                                          GAUSS130
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    GAUSS140
C                                                                          GAUSS150
      DOUBLE PRECISION SAVE1,SAVE2,SAVE3,SAVE4,SAVE5,STORE1,STORE2         GAUSS160
C                                                                          GAUSS170
      DIMENSION  CON(1), SAT(1), SDA(1), WRK(1), STT(1)                    GAUSS180
C                                                                          GAUSS190
      COMMON DATA                                                          GAUSS200
C                                                                          GAUSS210
      EQUIVALENCE  (DATA(  1),CON), (DATA( 16),SAT), (DATA( 36),SDA)       GAUSS220
     2             ,(DATA(286),STT), (DATA(391),WRK)                       GAUSS230
C                                                                          GAUSS240
      DIMENSION  R(3)      ,RD(3)     ,RDD(3)    ,COEF(8,3) ,SAV(3)        GAUSS250
C                                                                          GAUSS260
      EQUIVALENCE           (WRK( 52),R     ), (WRK( 55),RD    )           GAUSS270
     2  ,(WRK( 58),RDD   ), (WRK( 68),STEP  ), (WRK( 71),COEF  )           GAUSS280
C                                                                          GAUSS290
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *      GAUSS300
C                                                                          GAUSS310
C     UPDATE COEFFICIENTS * * * * * * * * * * * * * * * * * * * * * * *    GAUSS320
C                                                                          GAUSS330
      DO 1 I=1,3                                                           GAUSS340
      SAVE1 = COEF(3,I)                                                    GAUSS350
      SAVE2 = COEF(4,I)                                                    GAUSS360
      SAVE3 = COEF(5,I)                                                    GAUSS370
```

```
      SAVE4 = COEF(6,I)                                          GAUSS380
      SAVE5 = COEF(7,I)                                          GAUSS390
      COEF(2,I)= COEF(2,I) + ROD(I)                              GAUSS400
      COEF(1,I)= COEF(1,I) + COEF(2,I)                           GAUSS410
      STORE1 = ROD(I)                                            GAUSS420
      STORE2 = SAV(I)                                            GAUSS430
      COEF(3,I) = STORE1 - STORE2                                GAUSS440
      STORE1 = COEF(3,I)                                         GAUSS450
      COEF(4,I) = STORE1 - SAVE1                                 GAUSS460
      STORE1 = COEF(4,I)                                         GAUSS470
      COEF(5,I) = STORE1 - SAVE2                                 GAUSS480
      STORE1 = COEF(5,I)                                         GAUSS490
      COEF(6,I) = STORE1 - SAVE3                                 GAUSS500
      STORE1 = COEF(6,I)                                         GAUSS510
      COEF(7,I) = STORE1 - SAVE4                                 GAUSS520
      STORE1 = COEF(7,I)                                         GAUSS530
      COEF(8,I) = STORE1 - SAVE5                                 GAUSS540
    1 SAV(I) = ROD(I)                                            GAUSS550
C                                                                GAUSS560
C     STEP MOTION * * * * * * * * * * * * * * * * * * * * * * * * * GAUSS570
C                                                                GAUSS580
      H = STEP                                                   GAUSS590
      H2= H*H                                                    GAUSS600
      DO 10 I=1,3                                                GAUSS610
      R(I) = H2*( COEF(1,I) + ( ROD(I)+COEF(3,I))/12.            GAUSS620
     1          + ( 19.*COEF(4,I)+18.*COEF(5,I) )/240.           GAUSS630
     2          + ( 1726.*COEF(6,I)+1650.*COEF(7,I))/24192.      GAUSS640
     3          + 15.*COEF(8,I)/240.   )                         GAUSS650
      RD(I)= H *( COEF(2,I) + ( ROD(I)+COEF(3,I)*5./6.)/2.       GAUSS660
     1          + ( 3.*COEF(4,I)+251.*COEF(5,I)/90.)/8.          GAUSS670
     2          + 95.*COEF(6,I)/288. + 19087.*COEF(7,I)/60480.   GAUSS680
     3          + 5257.*COEF(8,I)/17280.  )                      GAUSS690
   10 CONTINUE                                                   GAUSS700
      RETURN                                                     GAUSS710
      END                                                        GAUSS720
```

STORAGE MAP

SUBROUTINE INTEG

COMMON VARIABLES

COMMON BLOCK // ORIGIN 00001 LENGTH 00744

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DATA	00000	R	DATA	00000	R	CON	00000	R
SAT	00017	R	SDA	00043	R	STT	00435	R
WRK	00606	R	R	00671	R	RD	00674	R
RDD	00677	R	STEP	00711	R	COEF	00714	R

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
SAVE1	00745	D	SAVE2	00747	D	SAVE3	00751	D
SAVE4	00753	D	SAVE5	00755	D	STORE1	00757	D
STORE2	00761	D	H	00763	R	H2	00764	R

ENTRY POINTS

INTEG SECTION 4

SUBROUTINES CALLED

SYSLOC SECTION 5

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1	29A	01121	10	55A	01252			

DECK LENGTH IN OCTAL IS 00337.

2.3.1.4 THE TRACKING GROUP

This group of routines is designed to define the position and velocity of the satellite relative to each of a set of prescribed tracking stations. This task is performed by defining the position vectors for the group of tracking stations as a function of the time, and associating with each, a set of unit vectors describing a topodetic coordinate system. This information is then utilized to determine the range, range-rate azimuth and elevation of the satellite relative to each of the stations.

The operations are performed with the aid of the following routines:

TRAK	Driver routine for computing the relative position and velocity data for each station
GHA	Defines the position of the Greenwich meridian as a function of universal time
UNIT	Constructs the position vector for the tracking station and the topodetic coordinate system
EQINOX	Computes the correction to the computational coordinate frame of 1950.0 (used for trajectory definition) to adjust for nutation and precession

Following the construction of the relative position information for a given station, a check is made to determine whether observation data are available for the epoch under analysis for that station. If not, the checking of the various stations is continued. However, if data are available, transfer is made from this group into the Filter group and a correction to the state vector is computed before continuing the station check process. This link between the two groups is very important and deserves considerable attention.

Subroutine TRAK

Purpose: TRAK is designed to check each of the tracking stations being employed at each point along the trajectory for the purpose of identifying those stations at which the vehicle is visible and to check the data tape to see if data is available at that time before transferring to the data filter.

Deck Name: TRACK

Calling Sequence: Call TRAK (ISTART)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
O	ISTART	-	1	Arg	Index utilized to restart the integration when the next data point has been read into memory
I	CONV1 CONV2	- -	1 1	CON (11) CON (12)	Conversion factors (days to seconds and degrees to radians)
I	NOUT	-	1	CON (14)	Output tape number
I	KCHECK	-	1	SAT (16)	Index utilized to check stations for visibility at all times or just at the times for which there is data
I	NOGO	-	1	SAT (17)	Index to limit number of stations checked
I	STATN	-	40	STA (1)	Array containing data for the stations to be checked (latitude, longitude, altitude and name)

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	HORCOR	Δ EI	10	STA (41)	Horizon corrections for each of the up to 10 stations employed
I	Number	-	1	STA (241)	Total number of stations being employed
I	ROTATE ROTINV	NP PTNT	3 x 3 3 x 3	WRK (1) WRK (10)	Matrix for transforming the frame of 1950.0 to that of data and its inverse. $\vec{r}_n = NP\vec{r}_{50}$
I	EN	N	3 x 3	WRK (19)	Nutation array relating the mean equator of date to the true equator of date. $\vec{r}_n = N\vec{r}_m$
I	TTRANW TTRANF	t_{n-1}	1 1	WRK (42) WRK (43)	Time of last data point
I	RVEC, VVEC	$\vec{r}_{50}$ $\vec{v}_{50}$	3 3	WRK (44) WRK (47)	The cartesian position and velocity vectors of the satellite at the time TW + TF in the reference frame of 1950.0
I	TW, TF	t_o	1,1	WRK (50) WRK (51)	The two words defining the epoch at which the station positions are desired (days) relative to 1950.0 (JD 2433282.423)
I	TWDATA TFDATA	t_n	1 1	WRK (61) WRK (62)	Time of next data point relative to the epoch of 1950.0
I	ITRAK	-	1	WRK (63)	Station which next senses the vehicle
O	H	h	1	WRK (68)	First approximation to step size for integration to the next time for which there is no data (sec)
I/O	RT	$\vec{r}_T$	3	WRK (95)	The tracking station position vector (Km) in the frame of date

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I/O	SLAT SLON SALT	L λ H	1 1 1	WRK (105) WRK (106) WRK (107)	Tracking station latitude, longitude and altitude (rad, rad, Km)
I/O	U E Z	U E N	3 3 3	WRK (108) WRK (124) WRK (127)	Up, east, north unit vectors at the tracking station being checked
O	RDATE VDATE	$\vec{r}_D$ $\vec{v}_D$	3 3	WRK (111) WRK (114)	The instantaneous position and velocity vectors in the true equator of date frame (Km, Km/sec)

Subroutines Required: ARKTNS (arc tangent)
CROSS (cross product)
FILTER (driver for filter chain)
GHA (Greenwich hour angle)
UNIT (Station vectors)
STMAT (Static transition matrix)
MATMPY (matrix multiplication)

Functions Required: AMAG (Vector magnitude)
ATAN (Arc tangent)
DOT (Dot product)

Approximate Deck Length: 650 (octal)

Description & Formulation:

TRAK is designed to check each of the stations being employed in the tracking network (one to ten) to determine the relative position of the vehicle at that epoch. As a portion of the process the range, range-rate, azimuth and elevation of the vehicle in topocentric coordinates are computed for the purpose of comparison with the actual tracking data in a differential correction of the trajectory itself. The procedure involved is presented below:

Upon defining the position and velocity vectors in the frame of date and computing the Greenwich Hour Angle associated with the time (GHA), the location of each of the stations is computed and a set of unit vectors defining a topocentric (radar Az-Eℓ) coordinate system is constructed (UNIT). At this point the relative position and velocity are computed

$$\vec{\rho} = \vec{r}_d - \vec{r}_t \quad ; \quad \rho = |\vec{\rho}| \quad ; \quad \dot{\rho} = \frac{\vec{\rho}}{\rho} \cdot (\vec{V}_d - \vec{V}_t)$$

and the elevation and azimuth are defined as follows:

$$\sin E\ell = \frac{\vec{\rho} \cdot \hat{U}}{\rho} \qquad -90 < E\ell < 90$$

$$\cos E\ell = +\sqrt{1-\sin^2 E\ell}$$

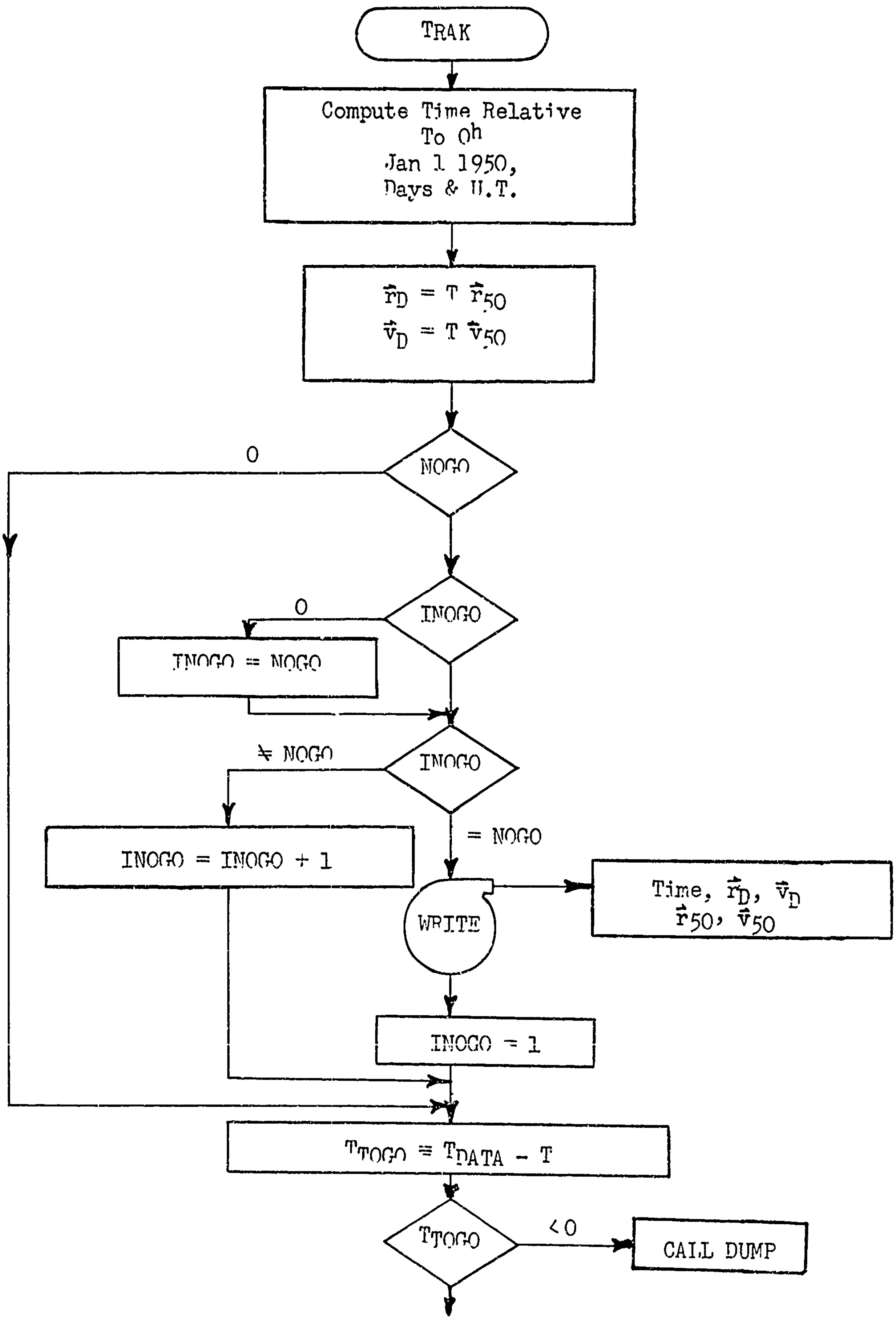
$$\cos a = \frac{\vec{\rho} \cdot \hat{N}}{\rho} = \cos A_z \cos E\ell$$

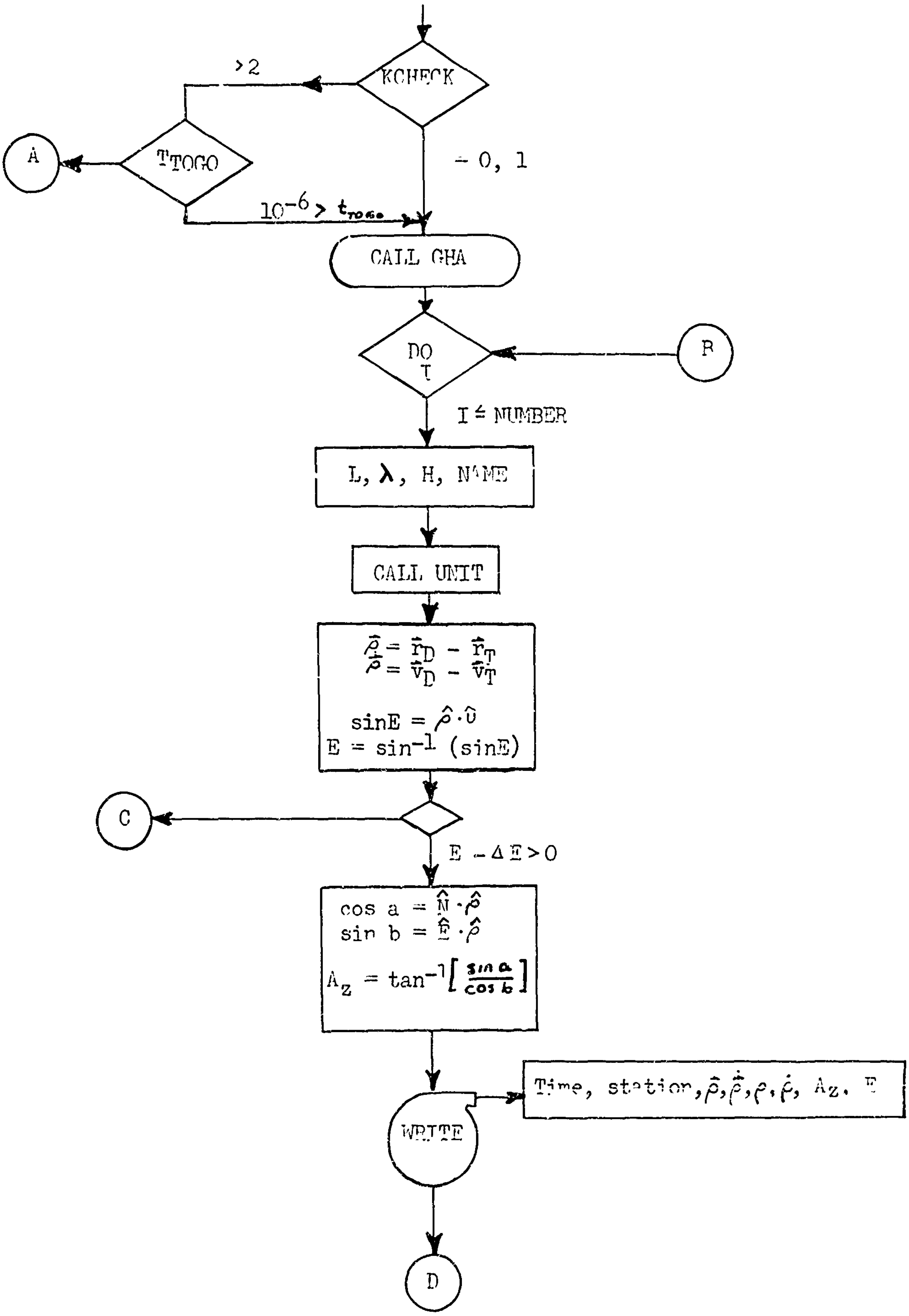
$$\sin b = \frac{\vec{\rho} \cdot \hat{E}}{\rho} = \sin A_z \cos E\ell$$

$$\tan A_z = \frac{\vec{\rho} \cdot \hat{E}}{\vec{\rho} \cdot \hat{N}}$$

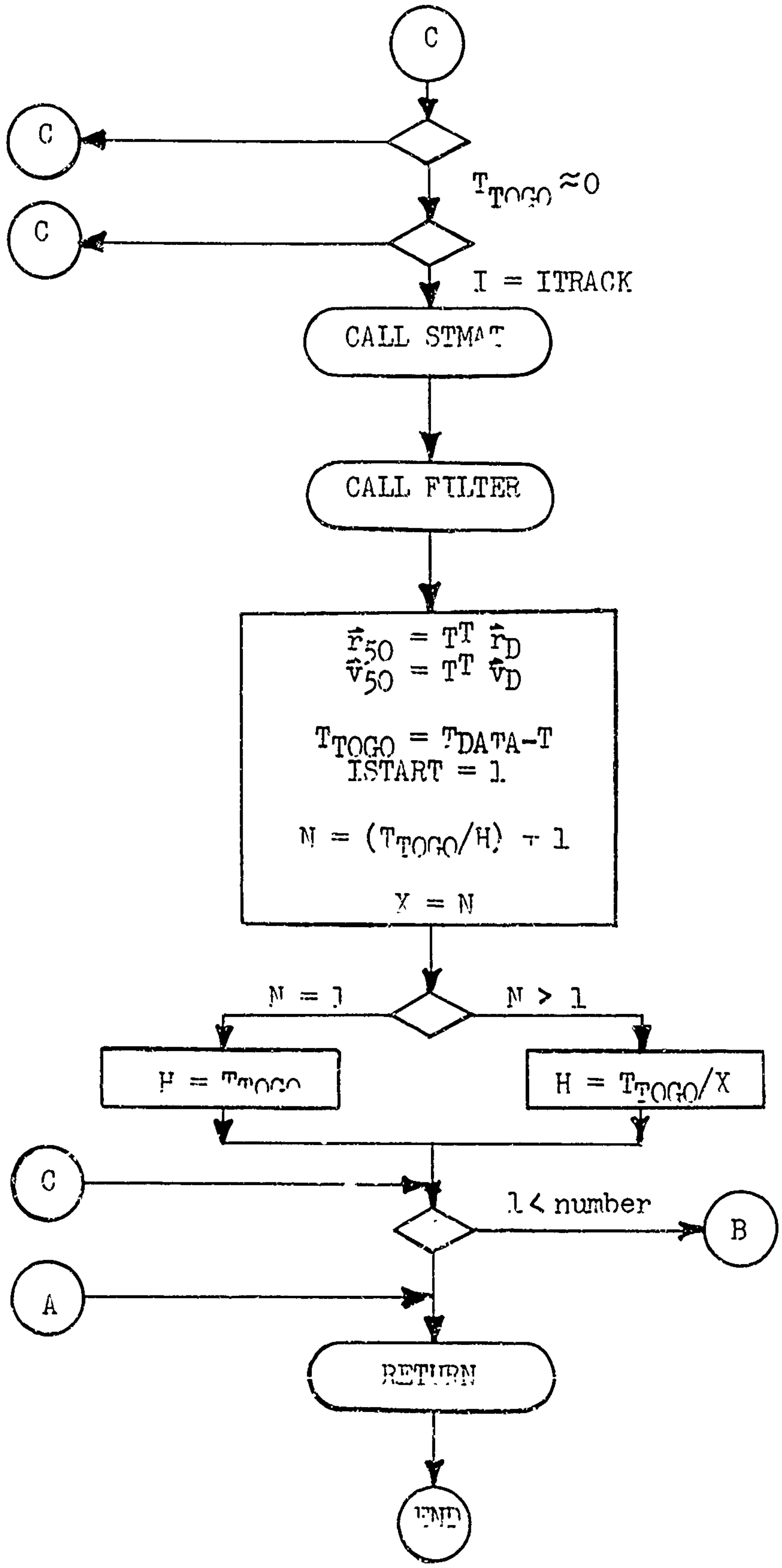
When the data point has been reached (i.e., the time at which data from the tracking stations is available) and associated with the proper station, the FILTER group is called and the data point processed. However, before the solution can proceed and before the trajectory can be stepped, the radius and velocity vectors in the frame of 1950.0 must be defined. This step is necessary due to the fact that subroutine UPSTAT may have adjusted the trajectory discretely in order to correct the tendency for the observed and computed trajectories to diverge.

TRAK is designed in such a manner that the routine can be by-passed in two distinctly different modes. First, if specified in the input data, the stations can be checked at every point along the trajectory (i.e., at every integration step) so as to provide a record of the track or only at those times when data is known to be available. This option drastically improves the general efficiency of the computational logic. Secondly, if desired, a minor efficiency can be effected in that only that station actually recording data at the time of the observation need be checked.


TRAK
Compute Time Relative To 0h Jan 1 1950, Days & U.T.
$\vec{r}_D = T\,\vec{r}_{50}$
$\vec{v}_D = T\,\vec{v}_{50}$
NOGO
0
INOGO
0
INOGO = NOGO
INOGO
≠ NOGO
= NOGO
INOGO = INOGO + 1
WRITE
Time, $\vec{r}_D$, $\vec{v}_D$
$\vec{r}_{50}$, $\vec{v}_{50}$
INOGO = 1
$T_{TOGO} = T_{DATA} - T$
T_{TOGO}
< 0
CALL DUMP



KCHECK
>2
TTOGO
A
= 0, 1
10^-6 > t_TOGO
CALL GHA
DO I
B
I ≤ NUMBER
L, λ, H, NAME
CALL UNIT
ρ = r_D - r_T
ρ̇ = v_D - v_T
sinE = ρ̂·Û
E = sin^-1 (sinE)
C
E - ΔE > 0
cos a = N̂·ρ̂
sin b = Ê·ρ̂
A_z = tan^-1 [sin a / cos b]
WRITE
Time, station, ρ, ρ̇, ρ, ρ̇, A_z, E
D


C
C
$T_{TOGO} \approx 0$
C
I = ITRACK
CALL FILTER
$\vec{r}_{50} = T^T \vec{r}_D$
$\vec{v}_{50} = T^T \vec{v}_D$
$T_{TOGO} = T_{DATA} - T$
ISTART = 1
X = N
N > 1
$H = T_{TOGO}/X$
C
B
A

```
      TRACK      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      SUBROUTINE TRAK(ISTART)                                           TRAK0020
C                                                                       TRAK0030
C     THIS ROUTINE IS DESIGNED TO DETERMINE WHICH OF THE SEVERAL        TRAK0040
C             TRACKING STATIONS UNDER CONSIDERATION IS CAPABLE OF       TRAK0050
C             OBSERVING THE VEHICLE AT ANY TIME , AND OF COMPUTING      TRAK0060
C             THE RANGE,RANG-RATE,AZIMUTH AND ELEVATION OF THE VEHICLE TRAK0070
C             RELATIVE TO ANY VISIBLE STATION . IF THE STATION OBSERVESTRAK0080
C             THE VEHICLE,ALL DATA ARE PRINTED OUT IN THE TOPOCENTRIC   TRAK0090
C             COORDINATE SYSTEM .                                       TRAK0100
C     IF A STATION IS VISIBLE,THE ROUTINE CHECKS TO SEE IF TRACKING     TRAK0110
C             DATA ARE AVAILABLE .IF SO,THE PROGRAM UTILIZES THE        TRAK0120
C             TRACKING DATA WHICH IT HAS AS THE COMPUTED VALUES OF THE TRAK0130
C             OBSERVATION VECTOR ,AND TRANSFERS INTO THE DATA FILTER . TRAK0140
C             UPON RETURN TO THIS ROUTINE , OPERATIONS WILL PROCEDE     TRAK0150
C             AS NORMAL( I.E. OTHER STATIONS WILL BE CHECKED ,THEN      TRAK0160
C             CONTROL WILL BE RELEASED FOR ANOTHER TRAJECTORY STEP ).   TRAK0170
C     THE OPTION IS PROVIDED TO BYPASS THIS CHECK PROCESS IF THE        TRAK0180
C             DATA POINT HAS NOT BEEN REACHED. THIS STEP WAS TAKEN IN   TRAK0190
C             THE INTEREST OF EFFICIENCY,BUT IT ALSO ALLOWS A SIMPLE    TRAK0200
C             MEANS OF SEPARATING THE TRUE DATA REDUCTION JOB FROM THE TRAK0210
C             TRAJECTORY PREDICTION JOB . THIS BYPASS IS TRIGGERED      TRAK0220
C             BY THE VARIABLE KCHECK WHICH IS OBTAINED FROM COMMON      TRAK0230
C             THROUGH THE INPUT DATA. A SECOND OPTION ALSO EXIST IF     TRAK0240
C             THE BYPASS IS USED BUT THE TIME HAS BEEN REACHED IN THAT TRAK0250
C             THE CHECK OF THE STATIONS WILL BE SHUNTED UNLESS THE      TRAK0260
C             STATION THAT RECORDED THE DATA IS THE STATION BEING       TRAK0270
C             CHECKED . THIS OPTION IS TRIGGERED BY NOGO (FROM INPUT)   TRAK0280
C     RVEC   = POSITION VECTOR  (1950.0)                                TRAK0290
C     VVEC   = VELOCITY VECTOR  (1950.0)                                TRAK0300
C     TW     = WHOLE NUMBER OF DAYS FROM 1950.0                         TRAK0310
C     TF     = FRACTIONAL PART OF A DAY DEFINING PRESENT EPOCH .        TRAK0320
C     KCHECK= 1 CHECK ALL STATIONS AT EACH STEP                         TRAK0330
C            = 2 CHECK STATIONS ONLY AT TIMES WHEN DATA IS AVAILABLE    TRAK0340
C     NOGO   = N WRITE OUT TRAJECTORY DATA EACH N-TH PASS (N=1,2...)    TRAK0350
C            = 0 SKIP OUTPUT OF TRAJECTORY DATA                         TRAK0360
C                                                                       TRAK0370
```

```
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * TRAK0380
C                                                                          TRAK0390
      DIMENSION  CON(1), SAT(1), SDA(1), WRK(1), STT(1)                   TRAK0400
C                                                                          TRAK0410
      COMMON DATA                                                          TRAK0420
C                                                                          TRAK0430
      EQUIVALENCE  (DATA(  1),CON), (DATA( 16),SAT), (DATA( 36),SDA)      TRAK0440
     2            ,(DATA(286),STT), (DATA(391),WRK)                        TRAK0450
C                                                                          TRAK0460
      DIMENSION  EN(3,3)   ,RVEC(3)    ,VVEC(3)   ,U(3)      ,ROTATE(3,3)TRAK0470
     2,F(3)      ,Z(3)      ,STATN(40)                                     TRAK0480
     3,RDATE(3)  ,VDATE(3)  ,RT(3)      ,RHO(3)    ,RUNIT(3)  ,HORCOR(10) TRAK0490
     4,SPIN(3)   ,VT(3)     ,VREL(3)                                       TRAK0500
C                                                                          TRAK0510
      EQUIVALENCE                                                          TRAK0520
     2   (CON( 11),CONV1 ), (CON( 12),CONV2 ), (CON( 14),NOUT  )          TRAK0530
     2  ,(SAT( 16),CODUMP), (SAT( 17),KCHECK), (SAT( 18),NOGO  )          TRAK0540
C                                                                          TRAK0550
      EQUIVALENCE             (WRK(   1),ROTATE), (WRK( 10),ROTINV)       TRAK0560
     2  ,(WRK( 19),EN     ), (WRK(111),RDATE ), (WRK(114),VDATE )         TRAK0570
     3  ,(WRK( 42),TTRANW), (WRK( 43),TTRANF), (WRK( 44),RVEC  )          TRAK0580
     4  ,(WRK( 47),VVEC  ), (WRK( 61),TWDATA), (WRK( 62),TFDATA)          TRAK0590
     5  ,(WRK( 63),ITRAK ), (WRK( 68),H     ), (WRK(105),SLAT  )          TRAK0600
     6  ,(WRK(106),SLON  ), (WRK(107),SALT  ), (WRK(108),U     )          TRAK0610
     7  ,(WRK(124),E     ), (WRK(127),Z     ), (WRK( 95),RT    )          TRAK0620
     8  ,(WRK( 69),TQ    ), (WRK( 70),TP    ), (WRK(134),N     )          TRAK0625
C                                                                          TRAK0630
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *     TRAK0640
C                                                                          TRAK0650
      EQUIVALENCE             (SDA(   1),STATN ), (SDA( 41),HORCOR)       TRAK0660
     2  ,(SDA(241),NUMBER) ,(WRK( 50),TW    ), (WRK( 51),TF    )          TRAK0670
C                                                                          TRAK0680
C                                                                          TRAK0690
      DATA  INOGO /0/                                                      TRAK0700
C                                                                          TRAK0710
C                                                                          TRAK0720
C     THE FIRST STEP IS TO CONVERT THE RADIUS AND VELOCITY VECTORS FOR    TRAK0730
```

```
C              THE EQUINOX OF 1950.0 TO THE FRAME OF THE TRUE EQUATOR   TRAK0740
C              OF DATE AND DEFINE THE POSITION AND VELOCITY VECTORS FOR TRAK0750
C              THE TRACKING STATIONS .                                  TRAK0760
      NM = TW                                                           TRAK0770
      TX = NM                                                           TRAK0780
      TY = (TF+(TW-TX))*86400.0                                         TRAK0790
      TP = TY-.077 *86400.                                              TRAK0791
      TQ = TX                                                           TRAK0792
      IF(TP) 399,398,393                                                TRAK0793
  399 TP = TP + 86400.                                                  TRAK0794
      TQ = TQ - 1.                                                      TRAK0795
  398 IPRINT = 1                                                        TRAK0800
      CALL MATMPY(ROTATE,3,3,RVEC,3,1,RDATE )                           TRAK0810    7
      CALL MATMPY( ROTATE,3,3,VVEC,3,1,VDATE )                          TRAK0820    9
  110 IF(NOGO.EQ.0)      GO TO 124                                      TRAK0830
  112 IF(INOGO.EQ.0)     INOGO = NOGO                                   TRAK0840
  114 IF(INOGO.NE.NOGO) GO TO 122                                       TRAK0850
      IPRINT = 0                                                        TRAK0860
  116 WRITE(    6,1030)  TQ,TP,RVEC,VVEC,RDATE,VDATE                    TRAK0910    19
 1030 FORMAT(1H-,24HTIME FROM JD 2433282.5 =, F17.8,3X,12HDAYS       ,2XTRAK0920
     2,F17.8,3X,10HSEC (U.T.)                                           TRAK0930
     3       /1H ,24HR VECTOR (1950)          =, 3E17.8                 TRAK0940
     4       /1H ,24HV VECTOR (1950)          =, 3E17.8                 TRAK0950
     5       /1H ,24HR VECTOR (DATE)          =, 3E17.8                 TRAK0960
     6       /1H ,24HV VECTOR (DATE)          =, 3E17.8 )               TRAK0970
  118 INOGO = 1                                                         TRAK0980
  120 GO TO 124                                                         TRAK0990
  122 INOGO = INOGO + 1                                                 TRAK1000
  124 CONTINUE                                                          TRAK1010
      TTOGO = ( TWDATA -TW )+(TFDATA-TF)                                TRAK1020
      IF(TTOGO.GE.0.) GO TO 800                                         TRAK1021
      WRITE(    6,875)                                                  TRAK1022    30
  875 FORMAT(68H1 TIME OF DATA POINT EXCEEDED , PROGRAM ERROR , EXECUTIOTRAK1023
     1N TERMINATED      )                                               TRAK1024
      CALL DUMP                                                         TRAK1025    31
  800 IF(KCHECK-2) 500,600,600                                          TRAK1030
  600 IF(ABS(TTOGO) - 1.E-6) 500,500,700                                TRAK1040
```

```
 500 SECOND = TP                                                   TRAK1050
     CALL GHA( SECOND,TO,GH,EN(2,1),OMEGA )                        TRAK1060    37
     OMEGA = OMEGA*CONV2                                           TRAK1070
     GH = GH * CONV2                                               TRAK1080
     DO 1000 I=1,NUMBER                                            TRAK1090
     J = 4*I-3                                                     TRAK1100
     SLAT =STATN(J)                                                TRAK1110
     SLON =STATN(J+1)                                              TRAK1120
     SALT =STATN(J+2)                                              TRAK1130
     SNAME=STATN(J+3)                                              TRAK1140
     CALL UNIT(GH)                                                 TRAK1150
C                                                                  TRAK1160
C    POSITION RELATIVE TO THE TRACKING STATION * * * * * * * * * * * * TRAK1170    47
     DO 1 K=1,3                                                    TRAK1180
   1 RHO(K) = RDATE(K) - RT(K)                                     TRAK1190
     RHOM = AMAG(RHO)                                              TRAK1200    56
     DO 2 K=1,3                                                    TRAK1210
   2 RUNIT(K) = RHO(K)/RHOM                                        TRAK1220
     SE = DOT(U,RUNIT)                                             TRAK1230    63
     IF(SE*SE-1.)  20,30,20                                        TRAK1240
  30 ELEV = 3.14159265* SIGN(.5,SE)                                TRAK1250
     GO TO 31                                                      TRAK1260
  20 ELEV = ATAN( SE/(SQRT(1.-SE**2)))                             TRAK1270    68     69
  31 ELEV = ELEV - HORCOR(I)                                       TRAK1275
     IF(ELEV)  555,4,4                                             TRAK1280
 555 IF(TTOGO - 1.E-6) 556,556,1000                                TRAK1290
 556 IF(I-ITRAK) 1000,4,1000                                       TRAK1300
   4 CAZ = DOT(RUNIT,Z)                                            TRAK1310    78
     SAZ = DOT(RUNIT,E)                                            TRAK1320    79
     AZMUTH =  ARKTNS (360,CAZ,SAZ)                                TRAK1330
C                                                                  TRAK1340
C    VELOCITY AND RANGE-RATE RELATIVE TO TRACKER * * * * * * * * * * TRAK1350    80
     SPIN(1)=0.                                                    TRAK1360
     SPIN(2)=0.                                                    TRAK1370
     SPIN(3)= OMEGA                                                TRAK1380
     CALL CROSS( SPIN,RT,VT )                                      TRAK1390    81
     DO 5 K =1,3                                                   TRAK1400
```

```
    5 VREL(K) = VVEC(K) - VT(K)                                          TRAK1410
      RHODOT= DOT(VREL,RUNIT )                                           TRAK1420    90
      ELED = ELEV/CONV2                                                  TRAK1430
      AZMUTD=AZMUTH/CONV2                                                TRAK1440
C                                                                        TRAK1450
C     OUTPUT OF DATA * * * * * * * * * * * * * * * * * * * * * * * * * * *TRAK1460
      IF( IPRINT -1 ) 1600,1500,1500                                     TRAK1470
 1500 WRITE(   6,1501) SNAME,TQ,TP                                       TRAK1480    92
 1501 FORMAT(/// 3X,A6,33H OBSERVES,TIME FROM JD 2433282.5=E17.8         TRAK1490
     1,3X,5HDAYS ,E17.8,3X,10HSEC(U.T.)      )                           TRAK1500
      IPRINT = 0                                                         TRAK1510
      GO TO 1502                                                         TRAK1520
 1600 WRITE(   6,1601) SNAME                                             TRAK1530    94
 1601 FORMAT(1H0,2X,A6 ,9H OBSERVES )                                    TRAK1540
 1502 WRITE(   6,10) (RHO(K),K=1,3),(VREL(K),K=1,3 ) , RHOM,RHODOT,      TRAK1550
     1               AZMUTD,ELED                                         TRAK1560    95
   10 FORMAT(1H ,4X,20HRELATIVE POSITION  = , 3E17.8                     TRAK1570
     3      /1H ,4X,20HRELATIVE VELOCITY  = , 3E17.8                     TRAK1580
     4      /1H ,4X,20HR, RDOT, AZ, ELEV  = , 4E17.8  )                  TRAK1590
C                                                                        TRAK1600
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  TRAK1610
C                                                                        TRAK1620
      IF(ABS(TTOGO)-1.E-6) 100,100,1000                                  TRAK1630
  100 IF(I-ITRAK) 1000,200,1000                                          TRAK1640
  200 TIME = CONV1*((TW-TTRANW)+(TF-TTRANF))                             TRAK1650
      CALL STMAT(TIME)                                                   TRAK1660    103
      CALL FILTER(I,RHOM,RHODOT,AZMUTH,ELEV)                             TRAK1670    110
      CALL MATMPY(R3TINV,3,3,RDATE,3,1,RVEC )                            TRAK1680    112
      CALL MATMPY(R3TINV,3,3,VDATE,3,1,VVEC )                            TRAK1690    114
      TTOGO =((TWDAT- TW) + (TFDATA-TF))*86400.                          TRAK1700
      ISTART = 1                                                         TRAK1710
      HNOM = 50.                                                         TRAK1720
      IF(H-HNOM) 926,926,927                                             TRAK1730
  927 HNOM = H                                                           TRAK1740
  926 N = TTOGO/HNOM + 1.                                                TRAK1750
      X = N                                                              TRAK1760
      IF(N-1) 752,752,753                                                TRAK1770
```

```
 752 H = TTOGO                                           TRAK1780
     GO TO 977                                           TRAK1790
 753 H = TTOGO/X                                         TRAK1800
 977 TTOGO = TTOGO/86400.                                TRAK1805
1000 CONTINUE                                            TRAK1810
 700 CONTINUE                                            TRAK1815
     RETURN                                              TRAK1820
     END                                                 TRAK1830
```

SUBROUTINE TRAK

COMMON VARIABLES

COMMON BLOCK // ORIGIN 00001 LENGTH 01014

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DATA	00000	R	DATA	00000	R	CON	00000	R
SAT	00017	R	SDA	00043	R	STT	00435	R
WRK	00606	R	CONV1	00012	R	CONV2	00013	R
NOUT	00015	I	CODUMP	00036	R	KCHECK	00037	I
NOGO	00040	I	ROTATE	00606	R	ROTINV	00617	R
EN	00630	R	RDATE	00764	R	VDATE	00767	R
TTRANW	00657	R	TTRANF	00660	R	PVEC	00661	R
VVFC	00664	R	TWDATA	00702	R	TFDATA	00703	R
ITRAK	00704	I	H	00711	R	SLAT	00756	R
SLON	00757	R	SALT	00760	R	U	00761	R
F	01001	R	Z	01004	R	RT	00744	R
TP	00712	R	TQ	00713	R	N	01013	I
STATN	00043	R	HORCOR	00113	R	NUMBER	00423	I
TW	00667	R	TF	00670	R			

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
RHO	01015	R	RUNIT	01020	R	SPIN	01023	R
VT	01026	R	VREL	01031	R			

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
NM	01034	I	TX	01035	R	TY	01036	R
IPRINT	01037	I	INDGO	01040	I	ITIGO	01041	R
SECOND	01042	R	GH	01043	R	OMEGA	01044	R

I	01045	I	J	01046	I	SNAME	01047	R
RHØM	01050	R	SE	01051	R	ELEV	01052	R
CAZ	01053	R	SAZ	01054	R	AZMUTH	01055	R
RHØDØT	01056	R	ELED	01057	R	AZMUTD	01060	R
TIME	01061	R	HNØM	01062	R	X	01063	R

ENTRY PØINTS

TRAK	SECTIØN	4

SUBRØUTINES CALLED

MATMPY	SECTIØN	5	.FWRD.	SECTIØN	6	.FSLØ.	SECTIØN	7
DUMP	SECTIØN	8	GHA	SECTIØN	9	UNIT	SECTIØN	10
AMAG	SECTIØN	11	DØT	SECTIØN	12	ATAN	SECTIØN	13
SQRT	SECTIØN	14	ARKTNS	SECTIØN	15	CRØSS	SECTIØN	16
STMAT	SECTIØN	17	FILTER	SECTIØN	18	.UN06.	SECTIØN	19
.FFIL.	SECTIØN	20	.FCNV.	SECTIØN	21	E.1	SECTIØN	22
E.2	SECTIØN	23	F.3	SECTIØN	24	E.4	SECTIØN	25
CC.1	SECTIØN	26	CC.2	SECTIØN	27	CC.3	SECTIØN	28
CC.4	SECTIØN	29	SYSLØC	SECTIØN	30			

EFN IFN CØRRESPØNDENCE

EFN	IFN	LØCATIØN	EFN	IFN	LØCATIØN	EFN	IFN	LØCATIØN
399	4A	01344	398	5A	01352	110	10A	01400
124	27A	01462	112	13A	01404	114	16A	01411
122	26A	01457	116	19A	01416	1030	FØRMAT	01107
118	24A	01454	120	25A	01456	800	32A	01507
875	FØRMAT	01205	500	36A	01520	600	34A	01513
700	129A	02205	1000	126A	02203	1	51A	01601
2	59A	01614	20	67A	01642	30	65A	01633
31	70A	01664	555	73A	01672	4	77A	01701
556	75A	01676	5	85A	01737	1600	94A	02002

****

TRACK STØRAGE MAP

1500	92A	01764	1501	FØRMAT	01223	1502	95A	02012
1601	FØRMAT	01244	10	FØRMAT	01251	100	105A	02047
200	107A	02052	926	119A	02146	927	118A	02144
752	122A	02172	753	124A	02175	977	125A	02200

DECK LENGTH IN ØCTAL IS 01214.

2.3.1.4.1 Subroutine UNIT

Purpose: UNIT computes the position vector defining the location of the tracking stations and evaluates the components of the (Up, East, North) unit vectors at the station.

Deck Name: UNIT

Calling Sequence: Call UNIT (GHA)

Input/Output

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	GHA	GHA	1	Arg	Hour Angle of Greenwich relative to true vernal equinox (rad)
I	RE	R_e	1	CON (1)	Equatorial radius of the earth (Km)
I	RPOL	R_p	1	CON (2)	Polar radius of the earth (Km)
O	RT	$\vec{r}_T$	3	WRK (95)	Position vector for the tracking station at this time (Km)
I	SLAT	L	1	WRK (105)	Station latitude (Geodetic) in radians
I	SLON	λ	1	WRK (106)	Station longitude in radians
I	SALT	H	1	WRK (107)	Station altitude (Km)
O	U,E,Z	U,E,N	3,3,3	WRK (108) WRK (124) WRK (127)	Up, East, North unit vectors expressed in cartesian coordinates in the true equator of date frame

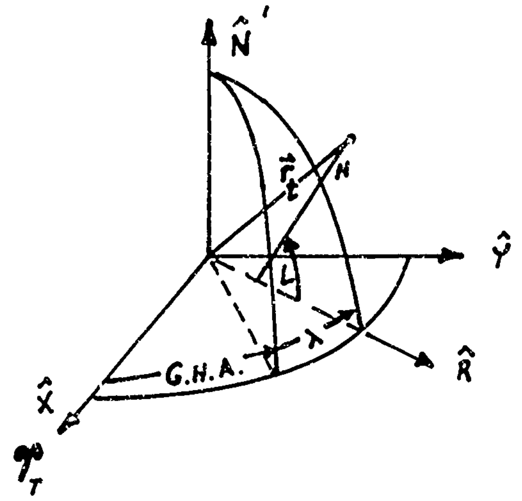
Subroutines Required: None

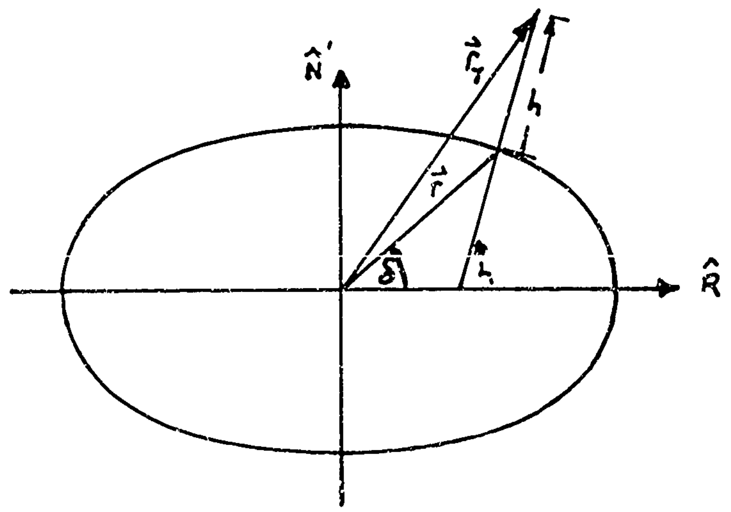
Functions Required: SIN (sine)
COS (cosine)
SQRT (square root)

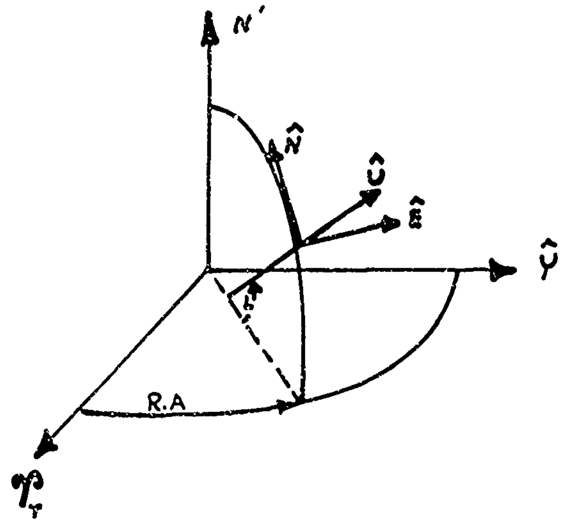
Approximate Deck Length: 170 (octal)

Formulation.

UNIT computes the components of a set of topocentric unit vectors for each of the tracking stations being checked as a function of the time and the radius vector of the tracking station itself. The unit vectors will be constructed first utilizing the information and definitions presented on the following sketches.







thus

$$\begin{bmatrix} U \\ E \\ N \end{bmatrix} = \begin{bmatrix} \cos L & 0 & \sin L \\ 0 & 1 & 0 \\ -\sin L & 0 & \cos L \end{bmatrix} \begin{bmatrix} \cos R.A. & \sin R.A. & 0 \\ -\sin R.A. & \cos R.A. & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Upsilon \\ Y \\ N' \end{bmatrix}$$

$$= \begin{bmatrix} \cos L \cos R.A. & \cos L \sin R.A. & \sin L \\ -\sin R.A. & \cos R.A. & 0 \\ -\sin L \cos R.A. & -\sin L \sin R.A. & \cos L \end{bmatrix} \begin{bmatrix} \Upsilon \\ Y \\ N' \end{bmatrix}$$

The vector defining the position of the tracking station is, however, more difficult to construct since it is not found by simple rotation. This construction is, however, simplified by noting that

$$\vec{r} = \vec{r} + h\,\hat{U}$$

$$= r\,(\cos\delta\,\hat{R} + \sin\delta\,\hat{N}') + h\,\hat{U}$$

and that the quantity δ can be evaluated by considering the equation of the ellipse in the $(\hat{R}, \hat{N}')$ plane

$$\frac{X^2}{a^2} + \frac{Z^2}{b^2} = 1$$

now

$$\frac{dZ}{dX} = -\frac{b^2}{a^2}\frac{X}{Z} = -\operatorname{Tan} L$$

and

$$\operatorname{Tan}\delta = \frac{Z}{X}$$

thus

$$\operatorname{Tan}\delta = \frac{b^2}{a^2}\tan L$$

But from the polar form of the equation of an ellipse

$$r = \frac{ab}{(a^2 \sin^2\delta + b^2\cos^2\delta)^{\frac{1}{2}}}$$

or

$$r \cos\delta = \frac{a}{\left(\frac{a^2}{b^2}\operatorname{Tan}^2\delta + 1\right)^{\frac{1}{2}}}$$

$$r \sin\delta = \frac{b}{\left(\frac{b^2}{a^2}\cot^2\delta + 1\right)^{\frac{1}{2}}}$$

so that upon substitution

$$\hat{r}_T = \left[\frac{a}{\left(\frac{b^2}{a^2}\sin^2 L + \cos^2 L\right)^{\frac{1}{2}}} + h\right](U_1 \hat{\Upsilon} + U_2 \hat{Y})$$

$$+ \left[\frac{b^2}{(b^2\sin^2 L + a^2\cos^2 L)^{\frac{1}{2}}} + h\right] U_3 \hat{N}'$$

where: a = equatorial radius (R_e)

b = polar radius (R_p)

Computational logic

UNIT

Unit Vectors

$$\hat{U} = \cos L \cos R.A.\ \hat{X} + \cos L \sin R.A.\ \hat{Y} + \sin L\ \hat{N}'$$
$$\hat{E} = -\sin R.A.\ \hat{X} + \cos R.A.\ \hat{Y} + 0\ \hat{N}'$$
$$\hat{N} = -\sin L \cos R.A.\ \hat{X} - \sin L \sin R.A.\ \hat{Y} + \cos L\ \hat{N}'$$

Tracking Station Position

$$\vec{r}_T = \left(\frac{R_e}{C} + h\right)\left(U_1 \hat{X} + U_2 \hat{Y}\right) + \left(\frac{R_p^2}{R_e C} + h\right)\left(U_3 \hat{N}'\right)$$

Return

END

```
      SUBROUTINE UNIT(GHA)                                              UNIT0020
C                                                                       UNIT0030
C     THIS ROUTINE COMPUTES THREE UNIT VECTORS WITH ORIGIN AT THE       UNIT0040
C              TRACKING STATION AND ORIENTED UP(U),EAST(E) AND NORTH(Z).UNIT0050
C              THE ROUTINE ALSO DEFINES THE POSITION VECTOR FOR THE     UNIT0060
C              STATION IN THE TRUE EQUATOR OF DATE FRAME (RT)           UNIT0070
C     SLAT = STATION LATITUDE ( RAD )                                   UNIT0080
C     SLON = STATION LONGITUDE( RAD )                                   UNIT0090
C     H    = STATION ALTITUDE (UNITS OF  POLAR OR EQUATORIAL RADII )    UNIT0100
C     RE   = EARTHS EQUATORIAL RADII                                    UNIT0110
C     RPOL = EARTHS POLAR RADII                                         UNIT0120
C                                                                       UNIT0130
      DIMENSION  CON(1), SAT(1), SDA(1), WRK(1), STT(1)                 UNIT0140
C                                                                       UNIT0150
      COMMON DATA                                                       UNIT0160
C                                                                       UNIT0170
      EQUIVALENCE  (DATA(  1),CON), (DATA( 16),SAT), (DATA( 36),SDA)    UNIT0180
     2            ,(DATA(286),STT), (DATA(391),WRK)                     UNIT0190
C                                                                       UNIT0200
      DIMENSION  RT(3)     ,U(3)     ,E(3)     ,Z(3)                    UNIT0210
C                                                                       UNIT0220
      EQUIVALENCE           (CON(  1),RE    ), (CON(  2),RPOL  )        UNIT0230
      EQUIVALENCE           (WRK( 95),RT    ), (WRK(105),SLAT  )        UNIT0240
     2  ,(WRK(106),SLON  ), (WRK(107),H     ), (WRK(108),U     )        UNIT0250
     3  ,(WRK(124),E     ), (WRK(127),Z     )                           UNIT0260
C                                                                       UNIT0270
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   UNIT0280
C                                                                       UNIT0290
      SLA  = SIN(SLAT)                                                  UNIT0300   2
      SLN  = SIN(SLON + GHA)                                            UNIT0310   3
      CLA  = COS(SLAT)                                                  UNIT0320   4
      CLN  = COS(SLON + GHA)                                            UNIT0330   5
      C = SQRT( CLA*CLA+((RPOL/RE)**2)*SLA*SLA)                         UNIT0340   6
      U(1) = CLA*CLN                                                    UNIT0350
      U(2) = CLA*SLN                                                    UNIT0360
      U(3) = SLA                                                        UNIT0370
```

```
      UNT          -  EFN    SCURCE STATEMENT  -  IFN(S)  -

   E(1) = -SLN                                                UNIT0380
   E(2) =  CLN                                                UNIT0390
   E(3) =  0.                                                 UNIT0400
   Z(1) =-SLA*CLN                                             UNIT0410
   Z(2) =-SLA*SLN                                             UNIT0420
   Z(3) = CLA                                                 UNIT0430
   RT(1)= (RE/C  +H)*U(1)                                     UNIT0440
   RT(2)= (RE/C  +H)*U(2)                                     UNIT0450
   RT(3)= ( RPOL*RPOL/(RE*C) +H )*U(3)                        UNIT0460
   RETURN                                                     UNIT047
   END                                                        UNIT048
```

STORAGE MAP

SUBROUTINE UNIT

COMMON VARIABLES

COMMON BLOCK // ORIGIN 00001 LENGTH 01007

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DATA	00000	R	DATA	00000	R	CON	00000	R
SAT	00017	R	SDA	00043	R	STT	00435	R
WRK	00606	R	RE	00000	R	RPOL	00001	R
RT	00744	R	SLAT	00756	R	SLON	00757	R
H	00740	R	J	00761	R	E	01001	R
Z	01004	R						

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
SLA	01010	R	SLN	01011	R	CLA	01012	R
CLN	01013	R	C	01014	R			

ENTRY POINTS

UNIT SECTION 4

SUBROUTINES CALLED

SIN	SECTION	5	COS	SECTION	6	SQRT	SECTION 7
SYSLOC	SECTION	8					

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION

DECK LENGTH IN OCTAL IS 00171.

2.3.1.4.2 Subroutine FQINØX

Purpose: EQINØX computes the transformation matrices relating the true equator of date frame of reference to the mean equator of 1950.0 (J.D. 2433282.423) frame.

Deck Name: ENØX

Calling Sequence: CALL EQINOX (TIME)

Input/Output

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	TIME	t	1	Arg	mean solar days since 1950.0 (J.D. 2433282.423)
O	ROTATE	NP	3 X 3	WRK (1)	rotation matrix transforming vectors in the frame of 1950.0 to the frame of date
O	ROTINV	$P^T N^T$	3 X 3	WRK (10)	rotational transformation to convert vectors in frame of date to frame of 1950.0
O	EN	N	3 X 3	WRK (19)	nutation matrix relating vectors in the mean equator of date frame to those in the frame of date (used in GHA)

Subroutines Required: MATMPY (matrix multiplication)

Functions Required: SIN (Sine)
COS (cosine)

Approximate Deck Length: 1540 (Octal)

Description and Formulation:

Over 2000 years ago it was discovered that the Vernal Equinox would move from east to west by 50". 2453 every year. This motion is called precession and is caused by the gravitational attraction of other celestial bodies acting on the equatorial bulge of the earth. If the earth were perfectly spherical and radially homogeneous, it would not experience any deviation from its mean equatorial pole. However, since the earth has an equatorial bulge, it experiences torques from the gravitational attraction of the sun and the moon. Due to the fact that the lunar orbital plane is approximately 5° oblique to the mean ecliptic, both the lunar and solar torques tend to align the equator with the ecliptic. The earth responds to this torque much like a spinning top responds to a torque. It precesses about the mean ecliptic pole. This precession is called luni-solar precession. Since the moon is so much closer to the earth than the sun, its contribution to luni-solar precession is approximately twice as much as that from the sun. The equatorial pole has an obliquity of about 23.5° so at the rate of precession mentioned earlier, the equatorial pole would very nearly trace a right circular cone every 25,800 years.

Just as the sun and moon cause the equatorial pole to precess, so do the planets of our solar system cause the ecliptic pole to precess; however, the magnitude of this planetary precession is very small and will be considered negligible in this discussion.

"Total general precession" is the sum of planetary and luni-solar precession and gives the changes in the mean vernal equinox of date from some epoch. Total general precession amounts to 50". 2453/year and can be considered uniform for practical use. This is the rate of westward rotation of the mean vernal equinox of date.

As the equatorial pole precesses about the ecliptic pole, it also experiences further disturbances known as nutations. Free Eulerian Nutations are those which would occur if the earth were simply set in rotation and left to itself without any disturbing forces. Forced nutations are those which are caused by the changing positions in space of the sun, earth, and moon, which in turn cause variations in their respective gravitational attractions to the earth.

The most significant nutation is the 19 Year Lunar Nutation. This nutation is caused by the precession of the moon's orbit, which is inclined about 5° oblique to the mean ecliptic. The line of nodes associated with these planes precesses with a period of about 18.6 years. The result is to change the direction of the small fluctuations in potential experienced by the earth-moon system.

Other forced nutations include the Semi-annual Solar Nutation and the Fortnightly Lunar Nutation. These phenomena are the result of the decreasing torque that the sun and moon apply to the earth as they approach the passing of the equatorial plane. Due to symmetry, the net torque, as one of these bodies passes through the equatorial plane, is zero.

The earth model that is generally used for the analvses of this motion is a rigid ellipsoid which is later simplified to an oblate spheroid. This model does not account for elasticity, fluidity and other physical properties of the earth; but it is sufficient to use for a fairly complete derivation of precession and nutation. It must thus be noted that the results of an analysis using such a simplified model are not exact.

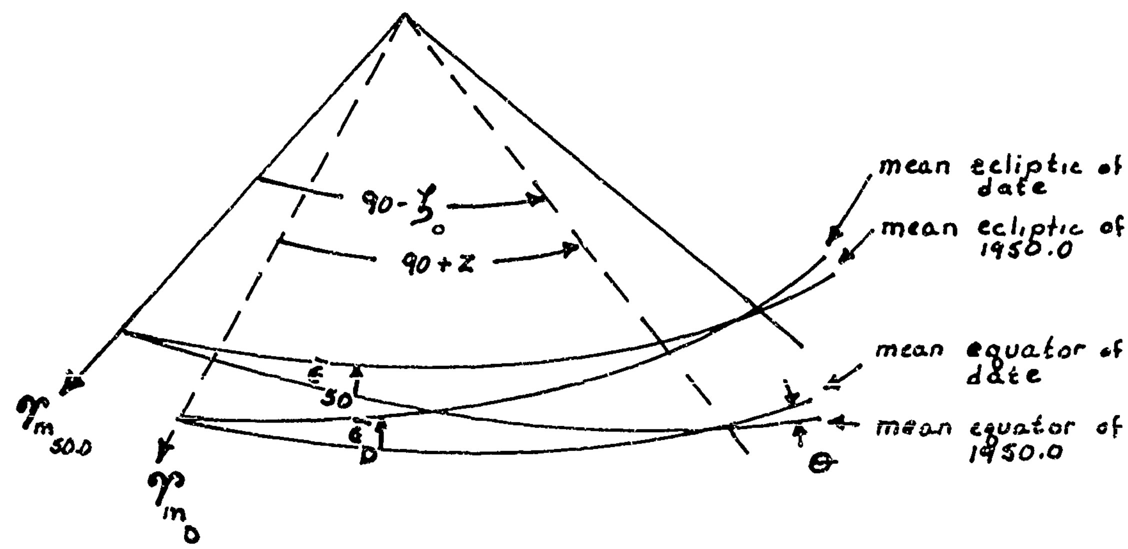
Improvement of the theoretical analysis resulting from the incorporation of observed data in the evaluation of the constants of integration and from the incorporation of more complete models of the earth in the analysis have, however, been effected. The results of these efforts based on formulations presented in the American Ephemeris and Nautical Almanac will be presented in the following paragraphs.

Precession:

Uniform precession is a rotation of the coordinate system defining the mean equator of date and may thus be represented by the matrix equation

$$\vec{r}_m = P\,\vec{r}_{50}$$

where $\vec{r}_{50}$ denotes the position vector in the standard inertial reference frame (fundamental plane and principal direction are the mean equator of zero hours, January 0, 1950 and the corresponding vernal equinox) and where $\vec{r}_m$ is the position vector in the mean equator of date frame. The problem thus, becomes one of determining the elements of P. This step in turn is accomplished by selecting 3 small parameters which relate the two frames. One such set is shown in the following sketch



The earth model that is generally used for the analyses of this motion is a rigid ellipsoid which is later simplified to an oblate spheroid. This model does not account for elasticity, fluidity and other physical properties of the earth; but it is sufficient to use for a fairly complete derivation of precession and nutation. It must thus be noted that the results of an analysis using such a simplified model are not exact.

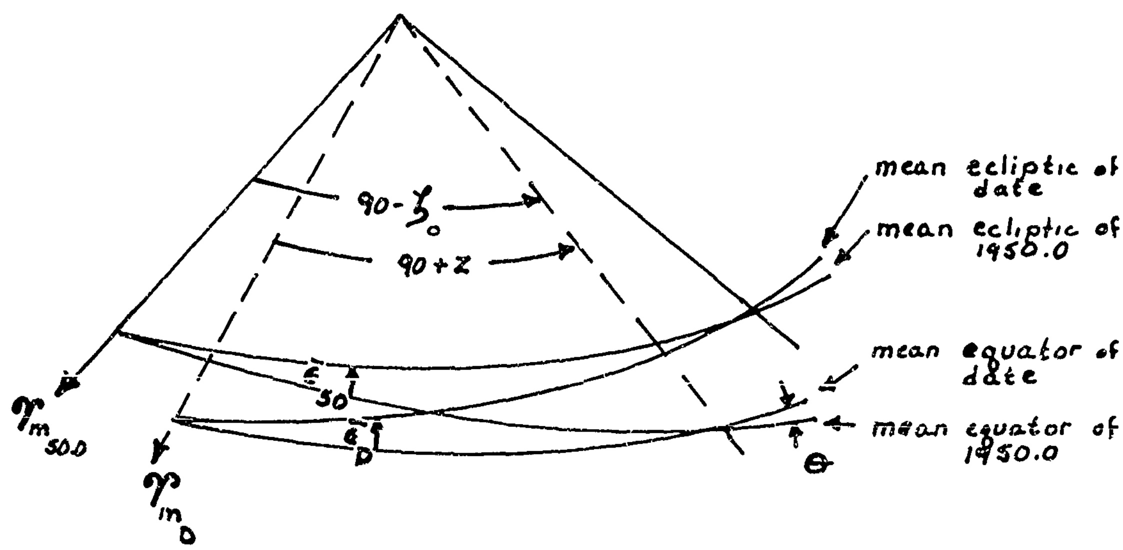
Improvement of the theoretical analysis resulting from the incorporation of observed data in the evaluation of the constants of integration and from the incorporation of more complete models of the earth in the analysis have, however, been effected. The results of these efforts based on formulations presented in the American Ephemeris and Nautical Almanac will be presented in the following paragraphs.

Precession:

Uniform precession is a rotation of the coordinate system defining the mean equator of date and may thus be represented by the matrix equation

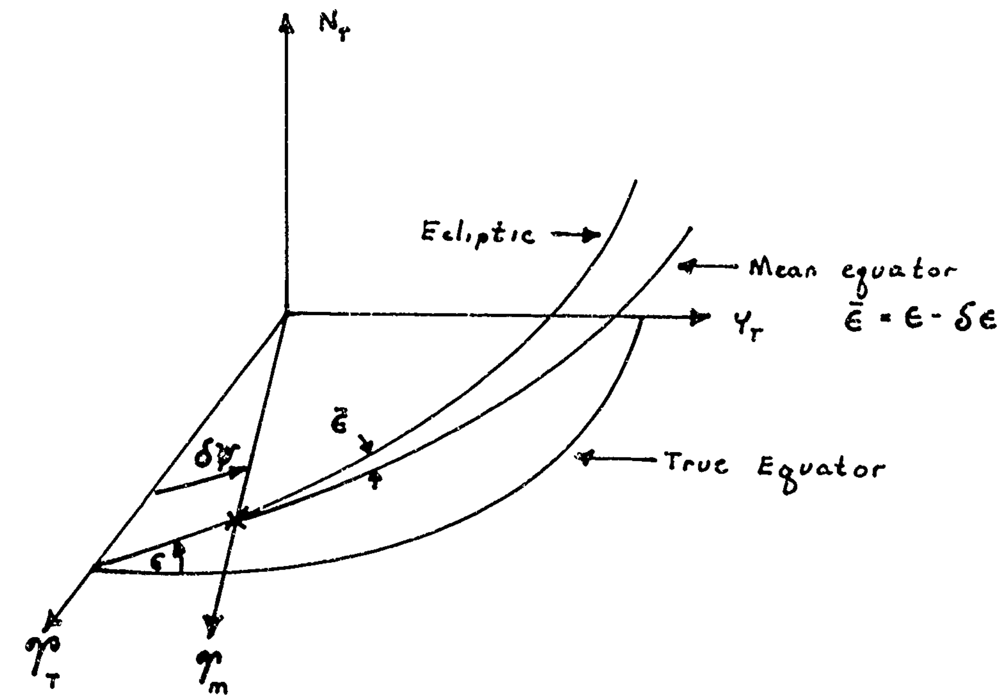
$$\vec{r}_m = P\,\vec{r}_{50}$$

where $\vec{r}_{50}$ denotes the position vector in the standard inertial reference frame (fundamental plane and principal direction are the mean equator of zero hours, January 0, 1950 and the corresponding vernal equinox) and where $\vec{r}_m$ is the position vector in the mean equator of date frame. The problem thus, becomes one of determining the elements of P. This step in turn is accomplished by selecting 3 small parameters which relate the two frames. One such set is shown in the following sketch



Nutation:

The relationship between the mean and true equator of date may be represented in terms of two small parameters as shown in the following sketch



and the small parameters ($\delta\psi$ and $\delta\epsilon$) can be divided into long ($\Delta\psi$ and $\Delta\epsilon$) and short ($d\psi$ and $d\epsilon$) period contributions which can be computed as a function of a set of quantities defined in the Nautical Almanac. These quantities are given in time series as:

$$\Omega = 12^\circ.112790 \quad -0^\circ.052953922\ D \quad +0^\circ.0020795T \quad +0^\circ.002081T^2 \quad +0^\circ.000002T^3$$

$$☾ = 64^\circ.375452 \quad +13^\circ.1763\text{97}\ D \quad -0^\circ.001131575T \quad -0^\circ.00113015T^2 \quad -0^\circ.0000019T^3$$

$$\Gamma' = 208^\circ.84399 \quad +0^\circ.11140408\ D \quad -0^\circ.010334T \quad -0^\circ.010343T^2 \quad -0^\circ.000012T^3$$

$$\Gamma = 282^\circ.08053 \quad +0^\circ.000047068\ D \quad +0^\circ.0004553T \quad +0^\circ.0004575T^2 \quad +0^\circ.000003T^3$$

$$L = 280^\circ.08121 \quad +0^\circ.98564734\ D \quad +0^\circ.000303\ (T+T^2)$$

where D = Days since reference epoch (1950.0) (J.D. 2433282.423)

T = Julian centuries past reference epoch

Corresponding to these time series the small parameters are:

$$\Delta\psi \times 10^4 = -(47^\circ\!.8927 + .0482T) \sin\Omega$$

$$+ .5800 \sin 2\Omega - 3.5361 \sin 2L - .1378 \sin (3L - \Gamma)$$

$$+ .0594 \sin (L + \Gamma) + .0344 \sin (2L - \Omega) + .0125 \sin (2\Gamma' - \Omega)$$

$$+ .3500 \sin (L - \Gamma) + .0125 \sin (2L - 2\Gamma')$$

$$d\psi \times 10^4 = -^\circ\!.5658 \sin 2☾ - .0950 \sin (2☾ - \Omega)$$

$$- .0725 \sin (3☾ - \Gamma') + .0317 \sin (☾ + \Gamma')$$

$$+ .0161 \sin (☾ - \Gamma' + \Omega) + .0158 \sin (☾ - \Gamma' - \Omega)$$

$$- .0144 \sin (3☾ + \Gamma' - 2L) - .0122 \sin (3☾ - \Gamma' - \Omega)$$

$$+ .1875 \sin (☾ - \Gamma') + .0078 \sin (2☾ - 2\Gamma')$$

$$+ .0414 \sin (☾ + \Gamma' - 2L) + .0167 \sin (2☾ - 2L)$$

$$- .0089 \sin (4☾ - 2L).$$

$$\Delta\epsilon \times 10^4 = 25^\circ\!.5844 \cos\Omega - .2511 \cos 2\Omega$$

$$+ 1.5336 \cos 2L + .0666 \cos (3L - \Gamma)$$

$$- .0258 \cos (L + \Gamma) - .0183 \cos (2L - \Omega)$$

$$- .0067 \cos (2\Gamma' - \Omega)$$

$$d\epsilon \times 10^4 = {}^\circ\!.2456 \cos 2☾ + .0508 \cos (2☾ - \Omega)$$

$$+ .0369 \cos (3☾ - \Gamma') - .0139 \cos (☾ + \Gamma')$$

$$- .0086 \cos (☾ - \Gamma' + \Omega) + .0083 \cos (☾ - \Gamma' - \Omega)$$

$$+ .0061 \cos (3☾ + \Gamma' - 2L) + .0064 \cos (3☾ - \Gamma' - \Omega)$$

Now since the mean obliquity of date is given by

$$\bar{\epsilon} = 23^\circ\!.4457587 - .01309404T - .00000088T^2 + .00000050T^3$$

and since

$$\epsilon = \bar{\epsilon} + \delta\epsilon$$

the rotational transformation relating the frames (X_m, Y_m, Z_m) and (X_T, Y_T, Z_T) is

$$\vec{r}_T = T_X(-\epsilon)\, T_Z(-\delta\psi)\, T_X(\bar{\epsilon})\, \vec{r}_m$$

$$\equiv N\, \vec{r}_m$$

where T ( ) denotes a rotational transformation, the subscript denotes the axis of rotation and the term in the parenthesis denotes the angle of rotation. Expansion of this transformation yields N as:

$n_{11} = \cos\delta\psi$

$n_{12} = -\sin\delta\psi\cos\bar{\epsilon}$

$n_{13} = -\sin\delta\psi\sin\bar{\epsilon}$

$n_{21} = \sin\delta\psi\cos\epsilon$

$n_{22} = \cos\delta\psi\cos\epsilon\cos\bar{\epsilon} + \sin\epsilon\sin\bar{\epsilon}$

$n_{23} = \cos\delta\psi\cos\epsilon\sin\bar{\epsilon} - \sin\epsilon\cos\bar{\epsilon}$

$n_{31} = \sin\delta\psi\sin\epsilon$

$n_{32} = \cos\delta\psi\sin\epsilon\cos\bar{\epsilon} - \cos\epsilon\sin\bar{\epsilon}$

$n_{33} = \cos\delta\psi\sin\epsilon\sin\bar{\epsilon} + \cos\epsilon\cos\bar{\epsilon}$

which upon substitution of $\bar{\epsilon} + \delta\epsilon$ for ϵ may be approximated as

$$N = \begin{bmatrix} 1 & -\delta\psi\cos\bar{\epsilon} & -\delta\psi\sin\bar{\epsilon} \\ \delta\psi\cos\bar{\epsilon} & 1 & -\delta\epsilon \\ \delta\psi\sin\bar{\epsilon} & \delta\epsilon & 1 \end{bmatrix}$$

Combined Nutation and Precession:

The true equator of data frame and that corresponding to the mean equator of 1950.0 can now be related by direct substitution of the results of the previous paragraphs.

$$\vec{r}_m = P\,\vec{r}_{50}$$

$$\vec{r}_T = N\,\vec{r}_m$$

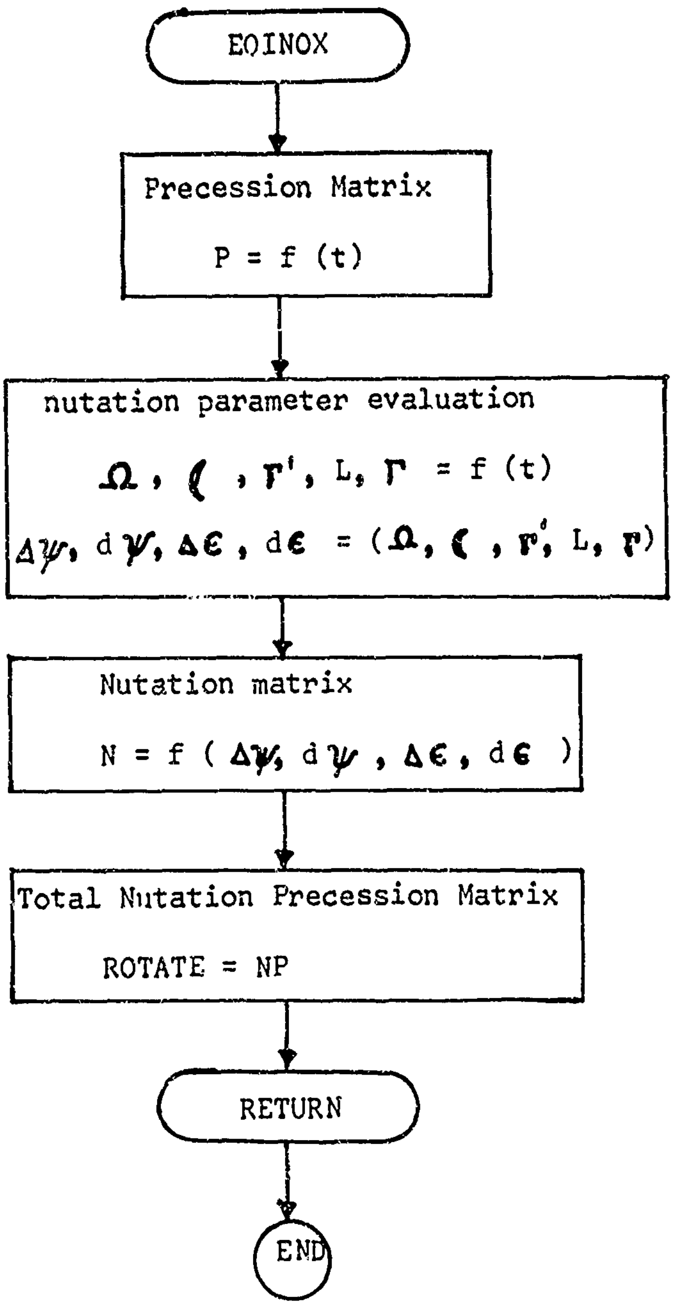
or

$$\vec{r}_T = NP\,\vec{r}_{50} \equiv [\text{ROTATE}]\,\vec{r}_{50}$$

and

$$\vec{r}_{50} = (NP)^T\,\vec{r}_D \equiv [\text{ROTINV}]\,\vec{r}_D$$

Computational Logic:



```
      SUBROUTINE EQINOX(TIME)                                              FNOX0020
C     THIS ROUTINE COMPUTES THE TRANSFORMATION MATRIX WHICH RELATES        FNOX0030
C             THE VECTORS IN THE COORDINATE FRAME OF 1950.0 TO THOSE       FNOX0040
C             IN THE TRUE EQUATOR OF DATE FRAME. THE ROUTINE ALSO          FNOX0050
C             STORES THE NUTATION MATRIX IN COMMON FOR OTHER USES.         FNOX0060
C             TIME IS DAYS FROM 1950.0.                                    FNOX0070
C                                                                          FNOX0080
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    FNOX0090
C                                                                          FNOX0100
      DIMENSION  CON(1), SAT(1), SDA(1), WRK(1), STT(1)                    FNOX0110
C                                                                          FNOX0120
      COMMON DATA                                                          FNOX0130
C                                                                          FNOX0140
      EQUIVALENCE  (DATA(  1),CON), (DATA( 16),SAT), (DATA( 36),SDA)       FNOX0150
     2            ,(DATA(286),STT), (DATA(391),WRK)                        FNOX0160
C                                                                          FNOX0170
C                                                                          FNOX0180
      DIMENSION  ROTATE(3,3),A(3,3),FN(3,3),ROTINV(3,3)                    FNOX0190
                                                                           FNOX0200
      EQUIVALENCE            (WRK(  1),ROTATE), (WRK( 10),ROTINV)          FNOX0210
     2                      ,(WRK( 19),FN     )                            FNOX0220
C                                                                          FNOX0230
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    FNOX0240
C                                                                          FNOX0250
C     THE FIRST STEP IS THE CONSTRUCTION OF THE MATRIX (A) WHICH DEFINESFNOX0260
C             THE EFFECTS OF PRECESSION ON THE MEAN EQUATOR OF DATE .      FNOX0270
      T = TIME/36525.                                                      FNOX0280
      T2 = T*T                                                             FNOX0290
      T3 = T2*T                                                            FNOX0300
      A(1,1) = 1. - .00029697*T2 - .00000013*T3                            FNOX0310
      A(1,2) = -.02234988*T - .00000676*T2 + .00000221*T3                  FNOX0320
      A(2,1) = -A(1,2)                                                     FNOX0330
      A(1,3) = -.00971711*T + .00000207*T2 + .00000096*T3                  FNOX0340
      A(3,1) = -A(1,3)                                                     FNOX0350
      A(2,2) = 1. - .00024976*T2 - .00000015*T3                            FNOX0360
      A(2,3) = -.00010859*T2 - .00000003*T3                                FNOX0370
```

```
   ****                                                          01/22/86                PAGE  6
      FNOX         -  FFN   SOURCE STATEMENT  -  IFN(S)  -

      A(3,2) = A(2,3)                                           FNOX0380
      A(3,3) = 1. - .00004721*T2 + .00000002*T3                 FNOX0390
C                                                               FNOX0400
C     THE SECOND STEP IS TO DEFINE THE EFFECT OF NUTATION OF THE EARTHS FNOX0410
C             SPIN AXIS AND THE DEPARTURES FROM THE MEAN FRAME OF DATE.FNOX0420
      D = TIME                                                  FNOX0430
      OM = 12.112790-.052953922*D+.0020795*T+.002081*T2+.000002*T3  FNOX0440
      CR = 64.375452+13.176397*D-.001131575*T-.00113015*T2+.0000019*T3  FNOX0450
      GP = 208.84399+.1114C408*D-.010334*T-.010343*T2-.000012*T3    FNOX0460
      VL = 280.08121+.98564734*D+.000303*(T+T2)                 FNOX0470
      G = 282.08053+.0000470684*D+.00045525*T+.0004575*T2+.000003*T3  FNOX0480
      OM = OM*.017453296                                        FNOX0490
      CR = CR*.017453296                                        FNOX0500
      GP = GP*.017453296                                        FNOX0510
      VL = VL*.017453296                                        FNOX0520
      G = G*.017453296                                          FNOX0530
      DE = 25.5844*COS (OM)-.2511*COS (2.*OM)+1.5336*COS (2.*VL)    FNOX0540
     1 +.0666*COS (3.*VL-G)-.0258*COS (VL+G)-.0183*COS (2.*VL-OM)   FNOX0550
     2 -.0067*COS (2.*GP-OM)                                    FNOX0560     2     3     4     5     6
                                                                             8
      DD = .2456*COS (2.*CR)+.0598*COS (2.*CR-OM)+.0369*COS (3.*CR-GP)  FNOX0570
     1 -.0139*COS (CR+GP)-.0086*COS (CR-GP+OM)+.0083*COS (CR-GP-OM)     FNOX0580
     2 +.0061*COS (3.*CR+GP-2.*VL)+.0064*COS (3.*CR-GP-OM)      FNOX0590     9    10    11    12     1
                                                                            15    16
      DT = -(47.8927+.0482*T)*SIN (OM)+.58*SIN (2.*OM)          FNOX0600
     1 -3.5361*SIN (2.*VL)-.1378*SIN (3.*VL-G)+.0594*SIN (VL+G)  FNOX0610
     2 +.0344*SIN (2.*VL-OM)+.0125*SIN (2.*GP-OM)+.35*SIN (VL-G) FNOX0620
     3 +.0125*SIN (2.*VL-2.*GP)                                 FNOX0630    17    18    19    20     2
                                                                            23    24    25
      DS = -.5658*SIN (2.*CR)-.095*SIN (2.*CR-OM)-.0725*SIN (3.*CR-GP)  FNOX0640
     1 +.0317*SIN (CR+GP)+.0161*SIN (CR-GP+OM)+.0158*SIN (CR-GP-OM)     FNOX0650
     2 -.0144*SIN (3.*CR+GP-2.*VL)-.0122*SIN (3.*CR-GP-OM)      FNOX0660
     3 +.1875*SIN (CR-GP)+.0078*SIN (2.*CR-2.*GP)               FNOX0670
     4 +.0414*SIN (CR+GP-2.*VL)+.0167*SIN (2.*CR-2.*VL)         FNOX0680
     5 -.0089*SIN (4.*CR-2.*VL)                                 FNOX0690    26    27    28    29     3
                                                                            32    33    34    35     3
                                                                            38
```

```
      DE = .17453296E-5*(DE+DD)                                  FNØX0700
      DT = .17453296E-5*(DT+DS)                                  FNØX0710
      EB = 23.4457587-.01309404*T-.00000088*T2+.0000005*T3       FNØX0720
      EB = EB*.017453296                                         FNØX0730
      EPSIL = EB+DE                                              FNØX0740
      EN(1,1) = 1.                                               FNØX0750
      EN(1,2) = -DT*COS (EB)                                     FNØX0760     39
      EN(1,3) = -DT*SIN (EB)                                     FNØX0770     40
      EN(2,1) = -EN(1,2)                                         FNØX0780
      EN(2,2) = 1.                                               FNØX0790
      EN(2,3) = -DE                                              FNØX0800
      EN(3,1) = -EN(1,3)                                         FNØX0810
      EN(3,2) = DE                                               FNØX0820
      EN(3,3) = 1.                                               FNØX0830
C                                                                FNØX0840
C     THE TOTAL ROTATION MATRIX IS NOW DEFINED BY UTILIZING      FNØX0850
C            MATRIX MULTIPLICATION.                              FNØX0860
      CALL MATMPY( EN,3,3,A,3,3,ROTATE)                          FNØX0870     41
      DO 100 I=1,3                                               FNØX0880
      DO 100 J=1,3                                               FNØX0890
  100 ROTINV(I,J)=ROTATE(J,I)                                    FNØX0900
      RETURN                                                     FNØX0910
      END                                                        FNØX0920
```

```
****
    ENOX                                   STORAGE  MAP                 01/22/86               PAGE  33

                                      SUBROUTINE  EQINOX

                                             COMMON   VARIABLES

                  COMMON BLOCK       //              ORIGIN       00001                LENGTH       00641

SYMBOL        LOCATION       TYPE    SYMBOL      LOCATION      TYPE      SYMBOL      LOCATION      TYPE
DATA           00000          R      DATA         00000         R        CON          00000         R
SAT            00017          R      SDA          00043         R        STT          00435         R
WRK            00606          R      ROTATE       00606         R        ROTINV       00617         R
EN             00630          R

                                           DIMENSIONED PROGRAM VARIABLES

SYMBOL        LOCATION       TYPE    SYMBOL      LOCATION      TYPE      SYMBOL      LOCATION      TYPE
A              00642          R

                                          UNDIMENSIONED PROGRAM VARIABLES

SYMBOL        LOCATION       TYPE    SYMBOL      LOCATION      TYPE      SYMBOL      LOCATION      TYPE
T              00653          R      T2           00654         R        T3           00655         R
D              00656          R      OM           00657         R        CP           00660         R
GP             00661          R      VL           00662         R        G            00663         R
DE             00664          R      DP           00665         R        DT           00666         R
DS             00667          R      EB           00670         R        EPSIL        00671         R
I              00672          I

                                                ENTRY POINTS

      EQINOX        SECTION      4

                                               SUBROUTINES CALLED
```

```
****                                                                    01/22/86                     PAGE  34
    ENMX                                              STØRAGE  MAP

  CØS          SECTIØN      5            SIN          SECTIØN     6           MATMPY       SECTIØN      7
  SYSLØC       SECTIØN      8

                                                     EFN    IFN    CØRRESPØNDENCE

EFN          IFN          LØCATIØN     EFN          IFN          LØCATIØN    EFN          IFN          LØCATIØN
100            49A          02343
  DECK LENGTH IN ØCTAL IS      01540.
```

2.3.1.4.3 Subroutine GHA

Purpose: GHA Computes the hour angle of the Greenwich meridian relative to the true vernal equinox of date in degrees.

Deck Name: GHAN

Calling Sequence: Call GHA (T, D, GH, DA, OMEGA)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	T	t	1	Arg	Universal time for the selected date at which GH is to be computed (sec.)
I	D	D_{50}	1	Arg	Mean solar days elapsed since 0^h Jan. 1, 1950 (integral number)
O	GH	$\Upsilon_T(t)$	1	Arg	Greenwich hour angle in degrees
I	DA	da	1	Arg	Nutation correction to reference GH to true equinox of date
O	OMEGA	ω	1	Arg	Rotational rate of the earth at the selected epoch

Subroutines Required: None

Functions Required: Sign

Approximate Deck Length: 160 (octal)

Formulation:

The hour angle of the Greenwich meridian relative to the mean vernal equinox of epoch T is given in the Nautical Almanac as

$$\gamma_m(t) = 100^\circ\!.07554260 + 0^\circ\!.985647346d + (2^\circ\!.9015) \times 10^{-13}d^2 + wt \pmod{360}$$

where d = the integral number of days past 0^h 1 January 1950

t = time in seconds past 0^h of the epoch data

$$\omega = \frac{.00417807417}{1 + (5.21)10^{-13}d}$$

Thus, in order to be consistent with the measure of time in the remaining portion of the program (time reckoned from 1950.0) it was simply required that the whole and fractional number of days (D_{50}) reckoned from 1950.0 be changed by the difference

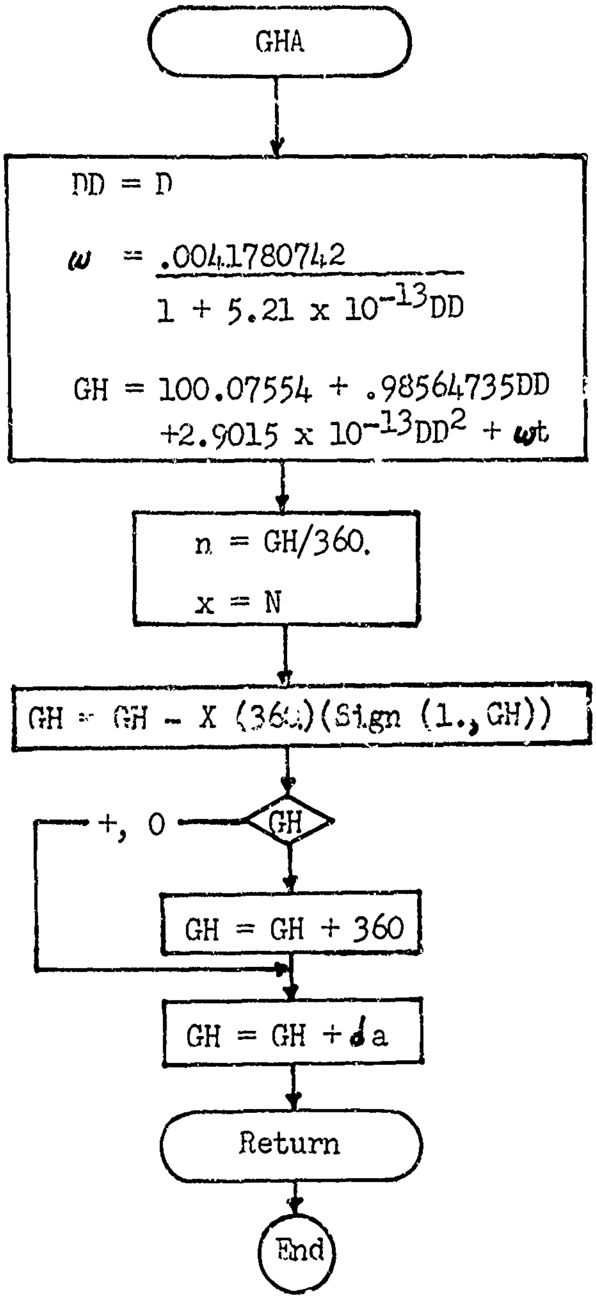
$$d = D_{50} - .077$$

and that a correction for nutation (da) be applied to compute the hour angle with respect to the true vernal equinox

$$da = \delta\psi \cos \bar{\epsilon}$$

$$\gamma_T(t) = \gamma_m(t) + da$$

Computational Logic:



```
      SUBROUTINE GHA(T,D,GH,DA,OMEGA )                                  GHA00020
C                                                                       GHA00030
C     GHA COMPUTES THE HOUR ANGLE OF GREENWICH RELATIVE TO THE MEAN     GHA00040
C              VERNAL EQUINOX OF DATE ( T IS UNIVERSAL TIME IN SECONDS. GHA00050
C              D IS DAYS SINCE ZERO HRS U.T.  1 JAN 1950 )              GHA00060
C                                                                       GHA00070
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   GHA00080
C                                                                       GHA00090
      DD=D                                                              GHA00100
      OMEGA = .0041780742/(1.+5.21E-13*DD )                             GHA00110
      GH = 100.07554 + .98564735*DD + 2.9015E-13*DD*DD + OMEGA*T        GHA00120
      N = GH/360.                                                       GHA00130
      X = N                                                             GHA00140
      GH = GH - X*360.*SIGN(1.,GH )                                     GHA00150
      IF(GH) 1,2,2                                                      GHA00160
    1 GH = GH + 360.                                                    GHA00170
    2 GH = GH + DA*57.295780                                            GHA00180
      RETURN                                                            GHA00190
      END                                                               GHA00200
```

SUBROUTINE GHA

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DD	00001	R	N	00002	I	X	00003	R

ENTRY POINTS

GHA	SECTION	3

SUBROUTINES CALLED

E.1	SECTION	4	E.2	SECTION	5	E.3	SECTION	6
E.4	SECTION	7	CC.1	SECTION	8	CC.2	SECTION	9
CC.3	SECTION	10	CC.4	SECTION	11	SYSLOC	SECTION	12

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1	4A	00106	2	5A	00111			

DECK LENGTH IN OCTAL IS 00156.

2.4 The Data Filter Group

The reduction of the data provided by the preprocessor (PROCES) will be accomplished utilizing a minimum variance recursive filter first developed by Dr. R. E. Kalman (the reference and a description of the filter will be presented in KALMAN). However, because there are several distinct operations involved in the process, it has been deemed desirable (from the standpoint of program modifications for other types of data, ease of development and checkout, etc.) to construct the filter in the form of several subroutines. This group of routines is the subject of the discussions which follow commencing with the driver (FILTER) and with the formulation (KALMAN). These routines are discussed prior to all of the remaining routines, since they establish the mathematical framework in which the others can best be understood, and because they establish the notation to be employed.

The interface of the differential corrections program and its preprocesson is also embedded in this group. This interface exists in the form of a routine (DATAPE) designed to read the specially prepared data tape and load the smoothed observation vector and identifying information into memory. The operation of this routine has been checked (as has the operation of all other routines in the program) to assure compatibility between the two programs.

2.4.1 Subroutine FILTER

Purpose: FILTER is designed as the driver for all routines in the Data Filter Group. It also serves the function of computing the observation vector (observed minus computed residuals).

Deck Name: FILT

Calling Sequence: CALL FILTER (NUMB, R, RD, A, E)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Description
I	NUMB	-	1	ARG	Number of the station being checked at this time in the routine TRAK
I	R RD A E	ρ $\dot{\rho}$ A E	1 1 1 1	ARG ARG ARG ARG	The computed values of range, range rate, azimuth and elevation relative to the tracking station (based) on the optimum estimate of the trajectory at this time (Km, Km/sec, rad, rad)
I	ODATA	-	3	WRK(65)	Observed values of range, range rate azimuth and/or elevation (no more than 3 pieces of this information at a time) (Km, Km/sec, rad, rad)
I	ITYPE	-	1	WRK(64)	The fixed point index which identifies which of the six possible types of data is being processed

Subroutines Required:

MEASURE	(computes $\frac{\partial \hat{y}}{\partial \vec{x}}$)
ERROR	(computes Q)
KALMAN	(computes $\vec{x}$ and P)
UPSTAT	(updates problem)
DATAPE	(provides next data point)
MAIN	(deck name of main program)

Functions Required: None

Approximate Deck Length: 205 (octal)

Description:

Subroutine FILTER serves as the driver routine for the complete filter package. It is designed to function in such a manner that all of the information required by KALMAN is available when the call is made. To be specific:

1) The first step made is the identification of the type of data being processed and the construction of the ordered set of observed minus computed residuals.

2) The next step is the computation of the matrix M^T (OBST = $\frac{\partial \vec{Y}}{\partial \vec{X}}^T$) in subroutine MEASUR and the definition of the weighting matrix in subroutine ERROR

3) At this point KALMAN is called and the state vector ($\overline{X}$) and the covariance matrix for the error in $\overline{X}$ are estimated.

4) UPSTAT is then entered to determine if the trajectory is known to a degree sufficient to allow-up-dating. If so, new conic elements are computed and written.

Upon return from UPSTAT the problem has been reduced by one of the smoothed data points (7 words - TW, TF, TYPE, STATION, OBSERVATION VECTOR (3)) and DATAPE is entered to determine if additional data are available. If so, the the next data point is read into memory and control is returned to the program. If there is no additional data, exit from the program is accomplished by calling the $IBFTC name of the main program (MAIN). This sequence is utilized at this time since it allows other cases to be run in a sequential manner, however, it is noted that a modification of this procedure may be required for other systems.

268

```
      SUBROUTINE FILTER(NUMB,R,RD,A,E)
C
C     FILTER SERVES AS THE DRIVER ROUTINE FOR THE DATA REDUCTION
C              PORTION OF THE PROGRAM. IT IS CALLED FROM SUBROUTINE
C              TRACK WHEN THE TIME IN DAYS IS EQUAL TO THE TIME AT
C              WHICH DATA IS AVAILABLE. THE FIRST STEP IS TO COMPUTE
C              THE OBSERVED*MINUS*COMPUTED RESIDUALS FOR THE DATA TYPE
C              SENSED AND SETUP THE FILTERING PROCESS.
C     ONCE THIS  STEP IS COMPLETE THE REMAINING ROUTINES IN THE FILTER
C              CHAIN CAN BE CALLED (MEASUR COMPUTES THE MATRIX OF
C              PARTIALS OF THE OBSERVABLES WITH RESPECT TO THE STATE ,
C              ERROR COMPUTES THE STATION AND NOISE ERROR MATRIX ,
C              TRANS COMPUTES THE STATE TRANSITION MATRIX , AND
C              KALMAN UPDATES THE STATE VECTOR AND THE ESTIMATION MATRIX
C     AT THIS POINT( PRIOR TO RETURN ) A CHECK IS MADE TO SEE IF THE
C             ERRORS IN THE ESTIMATION ARE SMALL COMPARED TO THE
C             TRUE POSITION AND VELOCITY . IF SO, THE TRAJECTORY
C             WILL BE ALTERED BEFORE PROCEDING TO THE NEXT STEP.(THE
C             ERROR SIGNAL,STATE,MUST ALSO BE ALTERED ).
C     NUMB  = NUMBER OF OBSERVING STATION
C     R,RD  = RANGE AND RANGE RATE RELATIVE TO STATION       (KM,KM/SEC)
C     A,E   = AZIMUTH AND ELEVATION IN TOPODETIC FRAME       (RAD)
C     KOUNT = 1 ** DATAPE RETURNS WITHOUT PERFORMING ANY FUNCTION
C           = 2 ** DATAPE READS THE NEXT OBSERVATION INTO MEMORY
C           = 3 ** WHEN KOUNT RETURNS A 3, THERE IS NO MORE DATA
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      DIMENSION  CON(1), SAT(1), SDA(1), WRK(1), STT(1)
C
      COMMON DATA
C
      EQUIVALENCE  (DATA(  1),CON), (DATA( 16),SAT), (DATA( 36),SDA)
     2             ,(DATA(286),STT), (DATA(391),WRK)
C
      DIMENSION  ODATA(3),  OBVEC(3)
```

STD 65 1203-1

```
C                                                                        FILT0380
      EQUIVALENCE            (WRK( 64),ITYPE ), (WRK( 65),ØDATA )         FILT0390
C                                                                        FILT0400
      DATA KØUNT/2/                                                      FILT0410
C                                                                        FILT0420
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  FILT0430
      DØ 1 I=1,3                                                         FILT0440
    1 ØBVEC(I) = 0.                                                      FILT0450
      GØ TØ(10,20,30,40,50,60),ITYPE                                     FILT0460
   10 ØBVEC(1)  =ØDATA( 1 ) - R                                          FILT0470
      GØ TØ 70                                                           FILT0480
   20 ØBVEC(1)  =ØDATA( 1 ) - RD                                         FILT0490
      GØ TØ 70                                                           FILT0500
   30 ØBVEC(1)  =ØDATA( 1 ) - A                                          FILT0510
      ØBVEC(2)  =ØDATA( 2 ) - E                                          FILT0520
      GØ TØ 70                                                           FILT0530
   40 ØBVEC(1)  =ØDATA( 1 ) - R                                          FILT0540
      ØBVEC(2)  =ØDATA( 2 ) - RD                                         FILT0550
      GØ TØ 70                                                           FILT0560
   50 ØBVEC(1) =ØDATA( 1 )  - R                                          FILT0570
      ØBVEC(2) =ØDATA( 2 )  - A                                          FILT0580
      ØBVEC(3) =ØDATA( 3 )  - E                                          FILT0590
      GØ TØ 70                                                           FILT0600
   60 ØBVEC(1) =ØDATA( 1 )  -RD                                          ****0610
      ØBVEC(2) =ØDATA( 2 )  -A                                           FILT0620
      ØBVEC(3) =ØDATA( 3 )  -E                                           FILT0630
   70 CALL MEASUR(M)                                                     FILT0640     22
      CALL ERRØR(NUMB,M)                                                 FILT0650     24
      CALL KALMAN(ØBVEC,M)                                               FILT0660     26
      CALL UPSTAT                                                        FILT0670     28
      CALL DATAPE(KØUNT)                                                 FILT0680     30
      IF( KØUNT.EQ.3 ) CALL MAIN                                         FILT0690     33
      RETURN                                                             FILT0700
      END                                                                FILT0710
```

```
****                                                             01/24/86                 PAGE  5
      FILT                                      STORAGE  MAP
                                          SUBROUTINE  FILTER

                                                COMMON   VARIABLES

               COMMON BLOCK        //              ORIGIN       00001            LENGTH       00711

SYMBOL     LOCATION     TYPE     SYMBOL     LOCATION     TYPE     SYMBOL     LOCATION     TYPE
DATA        00000        R       DATA        00000        R       CON         00000        R
SAT         00017        R       SDA         00043        R       STT         00435        R
WRK         00606        R       ITYPE       00705        I       ODATA       00706        R

                                           DIMENSIONED PROGRAM VARIABLES

SYMBOL     LOCATION     TYPE     SYMBOL     LOCATION     TYPE     SYMBOL     LOCATION     TYPE
OBVEC       00712        R

                                          UNDIMENSIONED PROGRAM VARIABLES

SYMBOL     LOCATION     TYPE     SYMBOL     LOCATION     TYPE     SYMBOL     LOCATION     TYPE
M           00715        I       KOUNT       00716        I

                                                ENTRY POINTS

    FILTER       SECTION      4

                                                SUBROUTINES CALLED

    MEASUR       SECTION      5          ERROR        SECTION      6          KALMAN       SECTION      7
    UPSTAT       SECTION      8          DATAPE       SECTION      9          MAIN         SECTION     10
    .FXEM.       SECTION     11          SYSLOC       SECTION     12
```

SID 65 1203-1

```
****
    FILT                                                         01/24/86              PAGE  6
                                                     STORAGE  MAP
                                            EFN   IFN   CORRESPONDENCE

EFN          IFN          LOCATION    EFN          IFN          LOCATION    EFN          IFN          LOCATION
1             5A           00726      10            10A          00 44      20            12A          00750
30           14A           00754      40            16A          00763      50            18A          00772
60           20A           01004      70            21A          01015
  DECK LENGTH IN OCTAL IS     00205.
```

2.4.1.1 Subroutine KALMAN

Purpose: To obtain the minimum variance estimate of the position and velocity relative to an estimated trajectory and to produce the covariance matrix for errors in the estimate.

Deck Name: KALM

Calling Sequence: Call KALMAN (OBVEC, M)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Description
I	OBVEC	Y	M	ARG	The vector composed of observed minus computed residuals (from FILTER)
I	M	-	1	ARG	The dimension of the observation vector
I	PHI	$\Phi(t,t_o)$	6 X 6	STT(1)	Transition matrix relating errors at two successive data points (from STMAT)
I	OBST	$M^T(t)$	6 X 3	STT(37)	Matrix of partials of observations with respect to the state (from MEASUR)
I	Q	$Q(t)$	3 X 3	STT(55)	Covariance matrix for contribution of errors in observations and of errors in station location (from ERROR)
I/O	STATE	$X(t)$	6	STT(64)	The vector $\left\{ \begin{matrix} \Delta\vec{r} \\ \Delta\vec{v} \end{matrix} \right\}$ where $\Delta\vec{r} = \vec{r} - \vec{r}_o$ $\Delta\vec{v} = \vec{v} - \vec{v}_o$ sub o = reference
I/O	P	$P(t)$	6 X 6	STT(70)	Covariance matrix for errors in the estimated state vector

Subroutines required:	TRANSP	(matrix transpose)
	MATMPY	(matrix multiplication)
	MTINY	(matrix inverse)
	SUBMAT	(matrix subtraction)
	ADDMAT	(matrix addition)
Functions required:	None	
Approximate Deck Length:	1056	(octal)

Formulation:

The theory of Kalman's recursive minimum variance data filter was first presented in a series of papers (e.g. Ref. 1) in which the author developed rigorously the form of the resultant estimate by employing an orthogonal projection lemma derived as a portion of the paper. Since this computational algorithm is utilized in the digital program being discussed, its form must be discussed. However, since complete descriptions of the development are recorded (Ref. 1), only a summary of the steps required will be presented.

The basic assumption of this procedure is that the optimal estimate of the state vector (X) for the system (in this case the vector $\left\{ \begin{smallmatrix} \Delta \bar{r} \\ \Delta \bar{v} \end{smallmatrix} \right\}$) is of the form. $\hat{X}^* = \sum_{i=1}^{n} a_i y_i$ where Y_i for this discussion are the components of the observed minus computed residuals ($Y = MX$), where the systems equations are

$$x(t) = \varphi(t, t_0)\ x(t_0) + u(t_0),$$

where $\varphi\ (t, t_0\ )$ is an n by n matrix of time-dependent coefficients and where u (t) is an independent Gaussian process **(u will not be utilized** in the estimation procedures). Lest this assumption be questioned on the grounds that it is excessively restrictive, note is made of the fact that Kalman proved that the results obtained with this model can, in general, be improved a with non-linear estimation only if the errors in the data and/or in the state vector for the system are non-Gaussian. Further, to achieve the improvement, at least third-order distribution functions for the errors are required.

Now, attention is turned to the problem of obtaining the optimum estimate of the state ($\hat{x}^*(t)$) at some time t by utilizing only the last estimate of the state ($\hat{x}^*(t-1)$) and the observation at time t. This statement of the problem leads to the basic form of the computation algorithm

$$\begin{aligned} \hat{x}^*(t) &= \varphi(t, t-1)\ \hat{x}^*(t-1) + K(t)\Big[Y(t) \\ &\quad - M(t)\,\varphi(t, t-1)\ \hat{x}^*(t-1)\Big] \\ &= \varphi(t, t-1)\ \hat{x}^*(t-1) + K(t)\,\Delta y(t) \end{aligned}$$

where the first term is the estimate of X(t) obtained utilizing data acquired prior to the last data point but propagated to time t. The second term is a weighted correction which is determined by the difference in the observed and predicted values of the observed minus computed residuals.

The task is now to define the optimum linear gain K(t) (optimum here will be taken to mean in the sense of minimum variance) to be utilized in the algorithm. This task will be accomplished by adopting the notation

$$y(t) = M(t)\,X(t) + \epsilon(t)$$

$$X(t) = \hat{X}^*(t) + \eta(t)$$

where ϵ (t) and η (t) are Gaussian errors and where the notation η (t) means the error at t based on all data processed prior to t. Expanding these identities

$$y(t) = M(t)\,\varphi(t,t-1)\,\hat{X}^*(t-1)$$

$$+ M(t)\,\varphi(t,t-1)\,\eta(t) + \epsilon(t)$$

$$\eta(t) = \left[\varphi(t,t-1) - K(t)\,M(t)\,\varphi(t,t-1)\right]\eta(t-1)$$

$$- K(t)\,\epsilon(t)$$

and noting that the optimum estimates for X satisfies the conditions that the covariance matrix for η (t) η^T(t) is minimized, or from the referenced lemma, that

$$E\left(\eta(t)\,Y^T(t)\right) = 0 \equiv \text{Expected value of } (\quad)$$

yields upon substitution and expansion (after the independence of η and $\tilde{X}$ and of E and $\tilde{X}$ are assumed).

$$\left[\varphi(t,t-1) - K(t)\,M(t)\,\varphi(t,t-1)\right] E\left[\eta(t-1)\,\eta^T(t-1)\right]\varphi^T(t,t-1)\,M^T(t)$$

$$- K(t)\,E\left[\epsilon(t)\,\epsilon^T(t)\right] = 0$$

At this point in the development, a slight change in the notation is adopted in that

$$P(t-1) \equiv E\left[\eta(t-1)\,\eta^T(t-1)\right]$$

$$Q(t) \equiv E\left[\epsilon(t)\,\epsilon^T(t)\right]$$

and it is noted that if P (t-1) is positive definite then the product

$$M(t)\varphi(t,t-1) \quad P(t-1)\varphi^T(t,t-1)M^T(t)$$

will also be positive definite provided that the observables (components of y) are linearly independent. Thus, the product is invertable; and the optimum weighting matrix is

$$K(t) = \varphi(t,t-1) \quad P(t-1) \quad \varphi^T(t,t-1)M^T(t)\cdot$$
$$\left[M(t)\varphi(t,t-1) \; P(t-1)\varphi^T(t,t-1)M^T(t) + Q(t)\right]^{-1}$$
$$\equiv P(t)M^T(t)\left[M(t)P(t)M^T(t) + Q(t)\right]^{-1}$$

Now since the distributions of the errors in the estimates can be prescribed at time zero and since the covariance matrix for the contribution of the stations to the uncertainty can be defined at any given time, the first weighting matrix can be computed. However, to reduce additional data points, the relationship defining the covariance matrix for the errors in the estimate after the reduction of the data point (P (t)) must be defined. This matrix may in turn be evaluated from the definition of P (t)

$$P^*(t) = E(\eta^*(t)\,\eta^{*T}(t))$$
$$= \varphi^*(t,t-1)\; P(t-1)\;\varphi^{*T}(t,t-1) + K(t)\;Q(t)\;K^T(t)$$

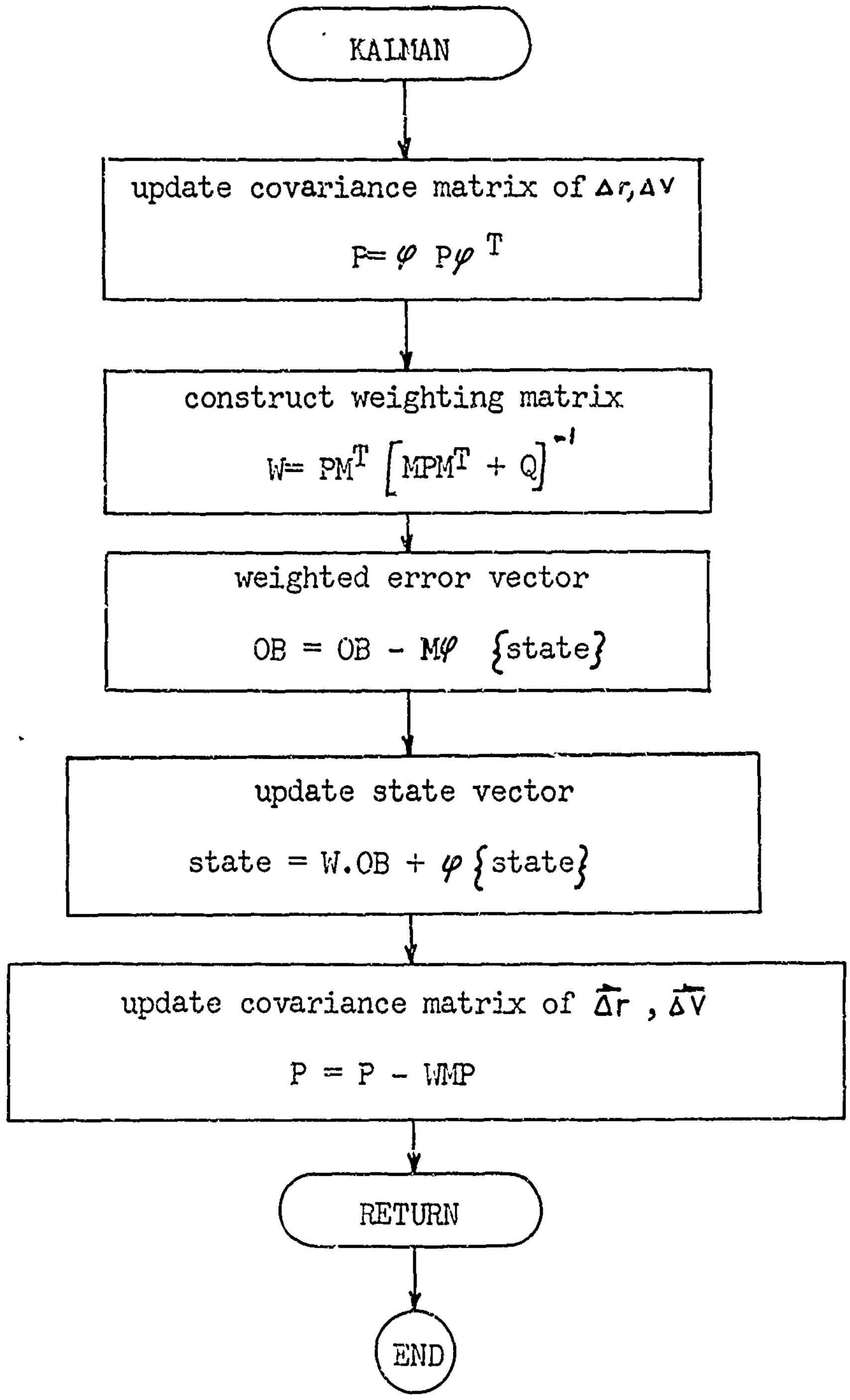
where $\varphi^*(t,t-1) = \varphi(t,t-1) - K(t)M(t)\,\varphi(t,t-1)$

Filter Description (operational characteristics)

The minimum variance formulation presented here is a simplification of the maximum likelihood estimator in that it assumes the errors are normally distributed. This, however, is the only valid objection voiced to the use of this procedure since in its general form the formulation includes most other data filters. (For example; weighted least squares - the simple case where the components of y are uncorrelated). The recursive nature of the filter also provides a distinct advantage since non-linear effects resulting from errors in the equations of motion and from non-precise relation of the errors at various points along the trajectory are minimized by the fact that time required for data accumulation is itself minimized. Further, if the trajectory is updated at each of the data points (assuming that the elements of P (t) are sufficiently small to make this practice feasible) the linear system can predict the behavior of the non-linear system to a very good degree. Finally, due to the recursive nature of the filter, the order of complexity in reducing any number of data points is constant; and problems of loss of numerical significance arising from manipulating large arrays of numbers are drastically reduced.

Ref. 1 Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems" Journal of Basic Engineering (March 1960.) pages 35-45

Computational Logic:



```
      SUBROUTINE KALMAN(OBVEC,M)                                        KALM0020
C                                                                       KALM0030
C     THIS ROUTINE COMPUTES THE KALMAN ESTIMATE OF THE STATE VECTOR     KALM0040
C             (I.E., THE DEVIATION IN THE SIX COMPONENTS OF POSITIONKALM0050
C             AND VELOCITY RELATIVE TO THE REFERENCE TRAJECTORY )       KALM0060
C             AND THE COVARIANCE MATRIX OF THE ERRORS IN THE            KALM0070
C             ESTIMATE . THIS FORMULATION IS BASED ON AN ORIGINAL       KALM0080
C             PAPER BY R.E. KALMAN (A NEW APPROACH TO LINEAR            KALM0090
C             FILTERING AND PREDICTION PROBLEMS ** JOURNAL OF BASIC     KALM0100
C             ENGINEERING ** MARCH 1960 ** PAGE 35-45 ) IN WHICH        KALM0110
C             A RECURSIVE LINEAR FILTER IS DISCUSSED                    KALM0120
C     STATE = 6-VECTOR OF DEPARTURES FROM REFERENCE TRAJECTORY AT THE   KALM0130
C             TIME THE LAST OBSERVATION DATA WERE PROCESSED . THIS      KALM0140
C             ESTIMATE IS UPDATED WITHIN THE ROUTINE                    KALM0150
C     OBVEC = M-VECTOR OF ERROR SIGNALS (OBSERVED MINUS COMPUTED ) AT   KALM0160
C             DISCRETE TIMES.                                           KALM0170
C     M     = DIMENSIONALITY OF OBVEC                                   KALM0180
C     PHI   = 6X6 STATE TRANSITION MATRIX (MATRIX OF PARTIALS           KALM0190
C             RELATING THE PRESENT STATE TO THE LAST KNOWN STATE )      KALM0200
C     OBS   = 3X6 MATRIX RELATING THE ERRORS IN THE OBSERVATION VECTOR  KALM0210
C             TO ERRORS IN THE STATE VECTOR                             KALM0220
C     P     = COVARIANCE MATRIX OF THE ERRORS IN THE ESTIMATED STATE    KALM0230
C             VECTOR AT THE TIME FOLLOWING THE LAST ESTIMATE OF         KALM0240
C             THE STATE.                                                KALM0250
C     Q     = COVARIANCE MATRIX FOR THE OBSERVATION ERRORS AT THE TIME  KALM0260
C             THE DATA BEING PROCESSED WAS COLLECTED .                  KALM0270
C     THE PROCEDURE WILL BE PERFORMED IN SEVERAL STEPS                  KALM0280
C             1**  COMPUTE WEIGHTING MATRIX                             KALM0290
C             2**  COMPUTE WEIGHTED TRANSITION MATRIX                   KALM0300
C             3**  ESTIMATE THE STATE VECTOR AT TIME ( T+1 )            KALM0310
C             4**  ESTIMATE THE COVARIANCE MATRIX FOR STATE ERRORS      KALM0320
C                                                                       KALM0330
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * KALM0340
C                                                                       KALM0350
      DIMENSION  CON(1), SAT(1), SDA(1), WRK(1), STT(1)                 KALM0360
C                                                                       KALM0370
```

```
     KALM        -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      COMMON DATA                                                        KALM0380
C                                                                        KALM0390
      EQUIVALENCE  (DATA(  1),CON), (DATA( 16),SAT), (DATA( 36),SDA)     KALM0400
     2            ,(DATA(286),STT), (DATA(391),WRK)                      KALM0410
C                                                                        KALM0420
      DIMENSION  STATE(6,1),P(6,6)    ,PHI(6,6)  ,PHIT(6,6) ,WPHI(6,6)   KALM0430
     2,DUM1(3,6),DUM2(3,6) ,DUM3(6,3) ,DUM4(6,3) ,DUM5(3,3) ,DUM6(3,3)   KALM0440
     3,DUM7(6,6),DUM8(6,6) ,DUM9(6,1) ,DUMX(6,1) ,DELTA(3,1),OBVEC(3,1)  KALM0450
     4,OBS(3,6) ,DUMMY(2,2),DUMY(2,2) ,WEIGHT(6,3),OBST(6,3),Q(3,3)      KALM0460
C                                                                        KALM0470
      EQUIVALENCE          (STT(  1),PHI   ), (STT( 37),OBST  )          KALM0480
     2  ,(STT( 55),Q     ), (STT( 64),STATE ), (STT( 70),P     )         KALM0490
C                                                                        KALM0500
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *      KALM0510
C                                                                        KALM0520
C     P-MATRIX AT TIME T * * * * * * * * * * * * * * * * * * * * * *     KALM0530
C                                                                        KALM0540
      CALL TRANSP(PHI,6,6,PHIT )                                         KALM0550     2
      CALL MATMPY(PHI,6,6,P,6,6,WPHI)                                    KALM0560     4
      CALL MATMPY(WPHI,6,6,PHIT,6,6,P)                                   KALM0570
C                                                                        KALM0580
C     WEIGHTING MATRIX   * * * * * * * * * * * * * * * * * * * * * *     KALM0590
C                                                                        KALM0600     6
      CALL TRANSP(OBST,6,3,OBS)                                          KALM0610     8
      CALL MATMPY(OBS,3,6,P,6,6,DUM1)                                    KALM0620     10
      CALL MATMPY(DUM1,3,6,OBST,6,3,DUM5 )                               KALM0630     12
      CALL ADDMAT(DUM5,3,3,Q,DUM5)                                       KALM0640     14
      DO 10 I=1,3                                                        KALM0650
      DO 10 J=1,3                                                        KALM0660
   10 DUM6(I,J) =0.                                                      KALM0670
      IF( M-2) 1,2,3                                                     KALM0680
    1 DUM6(1,1) = 1./ DUM5(1,1)                                          KALM0690
      GO TO 4                                                            KALM0700
    2 DO 11 I=1,2                                                        KALM0710
      DO 11 J=1,2                                                        KALM0720
   11 DUMMY(I,J) = DUM5(I,J)                                             KALM0730
      CALL MTINV( DUMMY,DUMY,2)                                          KALM0740     41
```

```
      DUM6(1,1) = DUMY(1,1)                                      KALM0750
      DUM6(1,2) = DUMY(1,2)                                      KALM0760
      DUM6(2,1) = DUMY(2,1)                                      KALM0770
      DUM6(2,2) = DUMY(2,2)                                      KALM0780
      GO TO 4                                                    KALM0790
    3 CALL MTINV( DUM5,DUM6,3 )                                  KALM0800   45
    4 CALL MATMPY(OBST,6,3,DUM6,3,3,DUM3 )                       KALM0810   47
      CALL MATMPY(P,6,6,DUM3,6,3,WEIGHT)                         KALM0820
C                                                                KALM0830
C     WEIGHTED OBSERVABLE VECTOR * * * * * * * * * * * * * * * * * * *  KALM0840
C                                                                KALM0850   49
      CALL MATMPY(OBS,3,6,PHI,6,6,DUM1)                          KALM0860   51
      CALL MATMPY(DUM1,3,6,STATE,6,1,DELTA )                     KALM0870   53
      CALL SUBMAT(OBVEC,3,1, DELTA,OBVEC )                       KALM0880
C                                                                KALM0890
C     UPDATE STATE VECTOR  * * * * * * * * * * * * * * * * * * * * *  KALM0900
C                                                                KALM0910   55
      CALL MATMPY( WEIGHT,6,3,OBVEC,3,1,DUM9)                    KALM0920   57
      CALL MATMPY(PHI,6,6,STATE,6,1,DUMX)                        KALM0930   59
      DO 5 I =1,6                                                KALM0940
    5 STATE(I,1)=DUMX(I,1)+ DUM9(I,1)                            KALM0950
C                                                                KALM0960
C     UPDATE COVARIANCE MATRIX (P)    * * * * * * * * * * .   * * * * *  KALM0970
C                                                                KALM0980
      CALL MATMPY(WEIGHT,6,3,OBS,3,6,WPHI)                       KALM0990   70
      CALL MATMPY(WPHI,6,6,P,6,6,DUM8)                           KALM1000   72
      CALL SUBMAT(P,6,6, DUM8,P)                                 KALM1010   74
      RETURN                                                     KALM1020
      END                                                        KALM1030
```

STØRAGE MAP

SUBRØUTINF KALMAN

COMMØN VARIABLES

COMMØN BLØCK // ØRIGIN 00001 LENGTH 00607

SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE
DATA	00000	R	DATA	00000	R	CØN	00000	R
SAT	00017	R	SDA	00043	R	STT	00435	R
WRK	00606	R	PHI	00435	R	ØBST	00501	R
Q	00523	R	STATE	00534	R	P	00542	R

DIMENSIØNED PRØGRAM VARIABLES

SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE
PHIT	00610	R	WPHI	00654	R	DUM1	00720	R
DUM2	00742	R	DUM3	00764	R	DUM4	01006	R
DUM5	01030	R	DUM6	01041	R	DUM7	01052	R
DUM8	01116	R	DUM9	01162	R	DUMX	01170	R
DELTA	01176	R	ØBS	01201	R	DUMMY	01223	R
DUMY	01227	R	WEIGHT	01233	R			

UNDIMENSIØNED PRØGRAM VARIABLES

SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPF
I	01255	I						

ENTRY PØINTS

KALMAN SECTIØN 4

SUBRØUTINES CALLED

KALM STORAGE MAP

TRANSP	SECTION	5	MATMPY	SECTION	6	ADDMAT	SECTION	7
MTINV	SECTION	8	SUBMAT	SECTION	9	SYSLOC	SECTION	10

EFN IFN CORRESPONDENCF

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
10	21A	01374	1	27A	01406	2	29A	01412
3	44A	01455	4	46A	01463	11	35A	01426
5	64A	01570						

DECK LENGTH IN OCTAL IS 01056.

2.4.1.2 Subroutine STMAT (State Transition Matrix)

Purpose: to provide the 6 X 6 matrix of partial derivatives o. the state vector at some arbitrary time with respect to the state vector at an earlier epoch.

Deck Name: STM

Calling Sequence: Call STMAT (TIME)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	RTRAN	$\vec{r}_1$	3	WRK (36)	radius vector in frame of date at time t_1
I	VTRAN	$\vec{V}_1$	3	WRK (39)	velocity vector in frame of date at time t_1
I	RDATE	$\vec{r}_2$	3	WRK (111)	radius at t_2
I	VDATE	$\vec{V}_2$	3	WRK (114)	velocity at t_2
I	TIME	Δt	1	ARG	$t_2 - t_1$
O	PHI	$\varphi(t_2,t_1)$	6 X 6	STT (1)	matrix $\left[\frac{\partial \vec{X}_2}{\partial \vec{X}_1}\right]$

Subroutines required: TRANS (conic Transition matrix)
INVAO (analytic transition inverse)

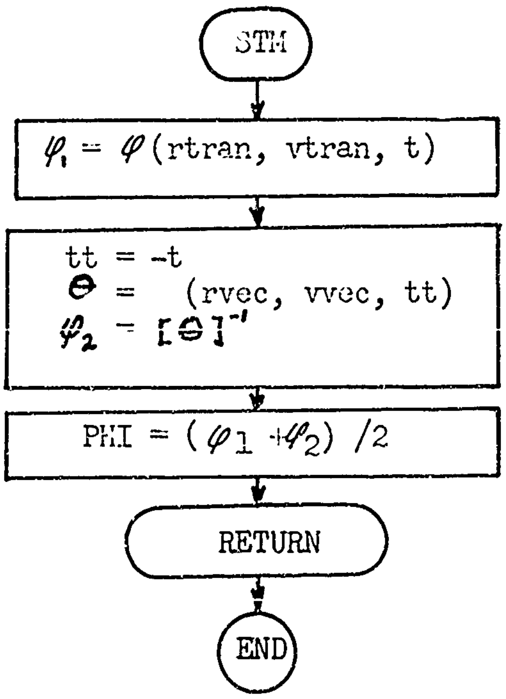
Functions required: None

Approximate Deck Length: 310 (octal)

Description:

The trajectories for the vehicles of interest to this study are nearly conic. Thus, partial derivatives evaluated for the nominal conic trajectory will agree well with those obtained from the true trajectory by more elaborate means such as the integration of the adjoint equations. For this reason, the conic representation will be utilized to construct the 6 X 6 matrix of partial derivatives though two slight modifications will be employed to assure that the conic and true matrices agree as well as possible. The first modification is that the position and velocity vectors used to define the partials will correspond to those of the true trajectory at the time from which the errors are propagating in this case, (the last data point) rather than the corresponding point on the conic references trajectory. The second modification is that the true position and velocity vectors at the time of the present data point will be utilized to obtain the inverse of a second estimate of this matrix by solving the conic problem backward in time. This second matrix will then be analytically inverted, utilizing a special property of this matrix (developed in INVAO) to obtain the desired partials. These two matrices of partials will not be identical because the conic trajectories utilized to define them differed in regard to each of the six constants of integration (the result of oblateness, drag --- forces). Further, they will differ from the true matrix. However, a first-order estimate of the effects of these forces can be included and the agreement with the true solution improved by averaging the two separate solutions for the partials.

Computational Logic:



```
      SUBROUTINE STMAT(TIME)                                           STM00020
C                                                                      STM00030
C     STMAT COMPUTES THE STATE TRANSITION MATRIX TO BE UTILIZED IN THE STM00040
C             FILTER . THIS IS ACCOMPLISHED IN AN APPROXIMATE SENSE    STM00050
C             BY CALLING THE TRANSITION MATRIX FOR CONIC MOTION TWICE  STM00060
C             (ONCE WITH R AND V AT THE LAST FILTER TIME AND ONCE AT   STM00070
C             THE PRESENT TIME ) AND AVERAGING THE RESULTS . THIS      STM00080
C             PROCESS ALLOWS FOR ORBITAL CHANGES DURING THE ELAPSED    STM00090
C             INTERVAL AND IS VERY ADEQUATE FOR MOST ORBITS .          STM00100
C                                                                      STM00110
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  STM00120
C                                                                      STM00130
      DIMENSION  CON(1), SAT(1), SDA(1), WRK(1), STT(1)                STM00140
C                                                                      STM00150
      COMMON DATA                                                      STM00160
C                                                                      STM00170
      EQUIVALENCE  (DATA(  1),CON), (DATA( 16),SAT), (DATA( 36),SDA)   STM00180
     2            ,(DATA(286),STT), (DATA(391),WRK)                    STM00190
C                                                                      STM00200
C                                                                      STM00210
      DIMENSION   RTRAN(3),VTRAN(3),RDATE(3),VDATE(3),P1(6,6),P2(6,6), STM00220
     1            PHI(6,6),R(3),V(3),Q(6,6)                            STM00230
C                                                                      STM00240
      EQUIVALENCE           (WRK( 36),RTRAN ), (WRK( 39),VTRAN )       STM00250
     2                     ,(WRK(111),RDATE ), (WRK(114),VDATE )       STM00260
     3                     ,(STT(  1),PHI   )                          STM00270
C                                                                      STM00280
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  STM00290
C                                                                      STM00300
      DO 1 I=1,3                                                       STM00310
      R(I) = RTRAN(I)                                                  STM00320
    1 V(I) = VTRAN(I)                                                  STM00330
      CALL TRANS(R,V,TIME,P1)                                          STM00340     12
      TT = -TIME                                                       STM00350
      DO 2 I=1,3                                                       STM00360
      R(I) = RDATE(I)                                                  STM00370
```

```
     STM          -  EFN   SOURCE STATEMENT  -  IFN(S)  -

  2 V(I) = VDATE(I)                                          STM00380
    CALL TRANS(R,V,TT,Q)                                     STM00390     24
    CALL INVAD(Q,P2)                                         STM00400     26
    DO 3 I=1,6                                               STM00410
    DO 3 J=1,6                                               STM00420
  3 PHI(I,J) = ( P1(I,J) + P2(I,J) )/2.                      STM00430
    RETURN                                                   STM00440
    END                                                      STM00450
```

**** STM | STORAGE MAP | 01/24/86 |

SUBROUTINE STMAT

COMMON VARIABLES

COMMON BLOCK // ORIGIN 00001 LENGTH 00772

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DATA	00000	R	DATA	00000	R	CON	00000	R
SAT	00017	R	SDA	00043	R	STT	00435	R
WRK	00606	R	RTRAN	00651	R	VTRAN	00654	R
RDATE	00764	R	VDATE	00767	R	PHI	00435	R

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
P1	00773	R	P2	01037	R	R	01103	R
V	01106	R	Q	01111	R			

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
I	01155	I	TT	01156	R			

ENTRY POINTS

STMAT SECTION 4

SUBROUTINES CALLED

TRANS SECTION 5 INVAD SECTION 6 SYSLOC SECTION 7

**** STM

STORAGE MAP

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1	7A	01167	2	19A	01207	3	33A	01245

DECK LENGTH IN OCTAL IS 00313.

2.4.1.2.1 Subroutine TRANS

Purpose: TRANS computes the matrix of partial derivatives of the state vector at an arbitrary point on a conic trajectory with respect to the state vector at another epoch. The variables utilized in this analysis are well defined for all conic motion.

Deck Name: TRAN

Calling Sequence: Call TRANS (R, V, T, PHI)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	R	$\vec{r}$	3	ARG	radius vector in cartesian coordinates at T = o (Km)
I	V	$\vec{v}$	3	ARG	velocity vector in cartesian coordinates at T = o (Km/sec)
I	T	Δt	1	ARG	time at which partials are desired (relative to point 1)
O	PHI	$\phi(t_2, t_1)$	6 X 6	ARG	matrix $\frac{\partial \bar{x}_2}{\partial \bar{x}}$
I	GM	μ	1	CON(6)	gravitational constant

Subroutines required: SEARCH (solve analog of Kepler's equation for position as a function of time)

Functions required:

AMAG (vector magnitude)
DOT (dot product)
SQRT (square root)
COS
SIN
COSH (hyperbolic sine)
SINH (hyperbolic cosine)
DERAQ ($\delta_{ij} = 0 \quad i \neq j$
$\qquad = 1 \quad i = j$)

Approximate Deck Length: 820 (decimal)

Formulation:

The equations of conic motion in terms of the so-called universal variables of Dr. S. Herrick will be utilized to develop the partial derivatives of the components of position and velocity at any given time (on the conic section) relative to the components of position and velocity at some other arbitrary epoch. This task will be performed utilizing a formulation valid for a non-rotating coordinate system and will be based on the development presented in the discussion of the reference trajectory. The required expressions for this analysis are:

$$\vec{r} = f\vec{r}_0 + g\vec{s}_0$$

$$\vec{s} = \dot{f}\vec{r}_0 + \dot{g}\vec{s}_0$$

where $f = 1 - \hat{C}/r_0$

$$g = \tau - \dot{U}$$

$$\dot{f} = -\hat{S}/r r_0$$

$$\dot{g} = 1 - \hat{C}/r$$

$\vec{r}$ = inertial position vector $= x\hat{x}^* + y\hat{y} + z\hat{z}$

$\vec{s}$ = normalized velocity vector $= \vec{V}/\sqrt{\mu} = \dot{x}\hat{x}^* + \dot{y}\hat{y} + \dot{z}\hat{z}$

$\hat{x}^*$ is the X unit vector. This notation is adopted to avoid confusion with a variable to be defined subsequently.

$$\hat{C} = a(1 - \cos\chi)$$

$$\hat{U} = a^{3/2}(\chi - \sin\chi)$$

$$\hat{S} = a^{1/2}(\sin\chi)$$

$$a = -1/\alpha$$

$$D_0 = \vec{r}_0 \cdot \vec{s}_0$$

$$C_0 = 1 + r_0\alpha$$

$$\chi = E - E_0 \qquad \text{(elliptic motion)}$$

$$\chi = F - F_0 \qquad (hyperbolic \quad motion)$$

$$\hat{\chi} = a^{1/2}\chi$$

The first step in obtaining the desired partial derivatives involves differentiation of the equations for $\bar{r}$ and $\bar{s}$ with respect to the components of $\bar{r}_0$ and $\bar{s}_0$. This task will be drastically simplified if full advantage is taken of the similar form of these derivatives at the outset. Thus, a shorthand notation will be adopted in that u and v ($\dot{u}$ and $\dot{v}$) can assume the values of s, y, and z ($\dot{x}$, $\dot{y}$, and $\dot{z}$)

$$\frac{\partial U}{\partial V_0} = f\,\delta_{UV} + U_0 \frac{\partial f}{\partial V_0} + \dot{U}_0 \frac{\partial g}{\partial V_0}$$

$$\frac{\partial U}{\partial \dot{V}} = g\,\delta_{UV} + U_0 \frac{\partial f}{\partial \dot{V}_0} + \dot{U}_0 \frac{\partial g}{\partial \dot{V}_0}$$

$$\frac{\partial \dot{U}}{\partial V_0} = \dot{f}\,\delta_{UV} + U_0 \frac{\partial \dot{f}}{\partial \dot{V}_0} + \dot{U}_0 \frac{\partial \dot{g}}{\partial \dot{V}_0}$$

$$\frac{\partial \dot{U}}{\partial \dot{V}_0} = \dot{g}\,\delta_{UV} + \frac{U_0\,\partial \dot{f}}{\partial \dot{V}_0} + U_0 \frac{\partial \dot{g}}{\partial \dot{V}_0}$$

where $\delta_{uv} = 0 \qquad u \neq v$

$\qquad\quad = 1 \qquad u = v$

Thus, the problem has reduced itself to one of obtaining the derivatives of f, g, $\dot{f}$ and $\dot{g}$ with respect to $\bar{r}$ and $\bar{s}$. This task will in turn be simplified if a set of intermediate parameters is selected, since the x...$\dot{s}$ do not appear explicitly in the equations for f...$\dot{g}$. The set to be utilized is suggested by the equation for the magnitude of r in this set of variables.

$$r = r_0 + D_0\,\hat{S} + (1 + r_0\,\alpha)\,\hat{C}$$

$$= F(r_0, D_0, \alpha)$$

Having selected the intermediate variables the next task is the differentiation of f, g, $\dot{f}$ and $\dot{g}$.

For f

$$\frac{\partial f}{\partial r_0} = \frac{-r_0 \dfrac{\partial \hat{C}}{\partial r_0} \quad r\hat{C}}{r_0^2}$$

$$\frac{\partial f}{\partial D_0} = \frac{-1}{r_0} \frac{\partial \hat{C}}{\partial D_0}$$

$$\frac{\partial f}{\partial \alpha} = -\frac{1}{r_0} \frac{\partial \hat{C}}{\partial \alpha}$$

For g

$$\frac{\partial g}{\partial r_0} = \frac{\partial \tau}{\partial r_0} - \frac{\partial \hat{U}}{\partial r_0}$$

$$\frac{\partial g}{\partial D_0} = \frac{\partial \tau}{\partial D_0} - \frac{\partial \hat{U}}{\partial D_0}$$

$$\frac{\partial g}{\partial \alpha} = \frac{\partial \tau}{\partial \alpha} - \frac{\partial \tilde{U}}{\partial \alpha}$$

For $\dot{f}$

$$\frac{\partial \dot{f}}{\partial r_0} = \frac{-(r r_0)\frac{\partial \hat{S}}{\partial r_0} + S\left(r + r_0 \frac{\partial r}{\partial r_0}\right)}{(r r_0)^2}$$

$$\frac{\partial \dot{f}}{\partial D_0} = \frac{-(r r_0)\frac{\partial \hat{S}}{\partial D_0} + \hat{S}\left(r_0 \frac{\partial r}{\partial D_0}\right)}{(r r_0)^2}$$

$$\frac{\partial \dot{f}}{\partial \alpha} = \frac{-(r r_0)\frac{\partial \hat{S}}{\partial \alpha} + \hat{S}\left(r_0 \frac{\partial r}{\partial \alpha}\right)}{(r r_0)^2}$$

and for $\dot{g}$

$$\frac{\partial \dot{g}}{\partial r_0} = \frac{-r\frac{\partial \hat{C}}{\partial r_0} + \hat{C}\frac{\partial r}{\partial r_0}}{r^2}$$

$$\frac{\partial \dot{g}}{\partial D_0} = \frac{-r\frac{\partial \hat{C}}{\partial D_0} + \hat{C}\frac{\partial r}{\partial D_0}}{r^2}$$

$$\frac{\partial \dot{g}}{\partial \alpha} = \frac{-r\frac{\partial \hat{C}}{\partial \alpha} + \hat{C}\frac{\partial r}{\partial \alpha}}{r^2}$$

Now attention turns to the derivatives of $\hat{C}$, $\hat{S}$, $\hat{U}$, τ etc. with respect to r_0, D_0, α.

for $\hat{C}$

$$\frac{\partial \hat{C}}{\partial r_0} = a \sin x \frac{\partial x}{\partial r_0} = \hat{S}\frac{\partial \hat{x}}{\partial r_0}$$

$$\frac{\partial \hat{C}}{\partial D_0} = \hat{S}\frac{\partial \hat{x}}{\partial D_0}$$

$$\frac{\partial \hat{C}}{\partial \alpha} = \hat{S}\frac{\partial \hat{x}}{\partial \alpha} + (1 - \cos x)\frac{\partial a}{\partial \alpha} = \hat{S}\frac{\partial \hat{x}}{\partial \alpha} + \hat{C}a$$

for $\hat{U}$ $$\frac{\partial \hat{U}}{\partial r_0} = a^{1/2}(1 - \cos x)\frac{\partial x}{\partial r_0} = \hat{C}\frac{\partial \hat{x}}{\partial r_0}$$

$$\frac{\partial \hat{U}}{\partial D_0} = \hat{C} \frac{\partial \hat{x}}{\partial D_0}$$

$$\frac{\partial \hat{U}}{\partial \alpha} = \hat{C} \frac{\partial \hat{x}}{\partial \alpha} + \frac{3}{2}(x - \sin x) a^{1/2} \frac{\partial a}{\partial \alpha}$$

$$= \hat{C} \frac{\partial \hat{x}}{\partial \alpha} + \frac{3}{2} \hat{U} a$$

for $\hat{S}$

$$\frac{\partial \hat{S}}{\partial r_0} = \cos x \frac{\partial \hat{x}}{\partial r_0}$$

$$\frac{\partial \hat{S}}{\partial D_0} = \cos x \frac{\partial \hat{x}}{\partial D_0}$$

$$\frac{\partial \hat{S}}{\partial \alpha} = \cos x \frac{\partial \hat{x}}{\partial \alpha} + \frac{1}{2} \sin x \, a^{3/2}$$

for τ

$$\frac{\partial \tau}{\partial r_0} = 0$$

$$\frac{\partial \tau}{\partial D_0} = 0$$

$$\frac{\partial \tau}{\partial \alpha} = \frac{1}{2} a^{3/2} t = \frac{1}{2} a \tau$$

and for r

$$\frac{\partial r}{\partial r_0} = 1 + D_0 \frac{\partial \hat{S}}{\partial r_0} + C_0 \frac{\partial \hat{C}}{\partial r_0} + \alpha \hat{C}$$

$$\frac{\partial r}{\partial D_0} = \hat{S} + D_0 \frac{\partial \hat{S}}{\partial D_0} + C_0 \frac{\partial \hat{C}}{\partial D_0}$$

$$\frac{\partial r}{\partial \alpha} = D_0 \frac{\partial \hat{S}}{\partial \alpha} + C_0 \frac{\partial \hat{C}}{\partial \alpha} + r_0 \hat{C} \equiv r_\alpha$$

The final set of partials required is now recognized to be that of with respect to r_0, and D_0, and α. This set requires the equation for time (analogous to Kepler's equation) be differentiated as follows:

$$\sqrt{a}\, t \equiv \tau = r_0 \hat{x} + D_0 \hat{C} + (1 + \alpha r_0) \hat{U}$$

for $\frac{\partial \hat{x}}{\partial r_0}$

$$0 = \hat{x} + r_0 \frac{\partial \hat{x}}{\partial r_0} + D_0 \frac{\partial \hat{C}}{\partial r_0} + \alpha \hat{U} + (1 + \alpha r_0) \frac{\partial \hat{U}}{\partial r_0}$$

$$= \hat{x} + \alpha \hat{U} + r \frac{\partial \hat{x}}{\partial r_0}$$

$$\frac{\partial \hat{\chi}}{\partial r_0} = -\frac{\hat{S}}{r}$$

for $0 = r_0 \frac{\partial \hat{\chi}}{\partial D_0} + \hat{C} + D_0 \frac{\partial \hat{C}}{\partial D_0} + C_0 \frac{\partial \hat{U}}{\partial D_0}$

$$\frac{\partial \hat{\chi}}{\partial D_0} = -\frac{\hat{C}}{r}$$

for $\frac{\partial \hat{\chi}}{\partial \alpha}$ $\frac{1}{2} a \Upsilon = r_0 \frac{\partial \hat{\chi}}{\partial \alpha} + D_0 \frac{\partial \hat{C}}{\partial \alpha} + C_0 \frac{\partial \hat{U}}{\partial \alpha} + \hat{U} r_0$

$$= \hat{U}(r_0 + 3/2\, C_0 a) + D_0(\hat{C}a) + \frac{\partial \hat{\chi}}{\partial \alpha}(r)$$

$$\frac{\partial \hat{\chi}}{\partial \alpha} = -\frac{1}{r}\left[\hat{U}(C_0 a + r_0) + \frac{a}{2}\left(D_0 \hat{C} - r_0 \hat{\chi}\right)\right] \equiv -\frac{\chi_\alpha}{r}$$

Now, substituting back into the previous expressions and collecting terms, the derivatives required to compute the partials of f, g, $\dot{f}$, $\dot{g}$, with respect to the intermediate parameters are:

for $\hat{C}$

$$\frac{\partial \hat{C}}{\partial r_0} = -\frac{\hat{S}^2}{r}$$

$$\frac{\partial \hat{C}}{\partial D_0} = -\frac{\hat{S}\hat{C}}{r}$$

$$\frac{\partial \hat{C}}{\partial \alpha} = -\frac{\hat{S}}{r}\chi + \hat{C}a \equiv C_\alpha$$

for $\hat{U}$

$$\frac{\partial \hat{U}}{\partial r_0} = -\frac{\hat{C}\hat{S}}{r}$$

$$\frac{\partial \hat{U}}{\partial D_0} = -\frac{\hat{C}^2}{r}$$

$$\frac{\partial \hat{U}}{\partial \alpha} = -\frac{\hat{C}\chi}{r} + \frac{3}{2}\hat{U}a \equiv U_\alpha$$

for $\hat{S}$

$$\frac{\partial \hat{S}}{\partial r_0} = -\cos\chi \frac{\hat{S}}{r}$$

$$\frac{\partial \hat{S}}{\partial D_0} = -\cos\chi \frac{\hat{C}}{r}$$

$$\frac{\partial \hat{S}}{\partial \alpha} = \cos\chi\, \chi_\alpha + \frac{1}{2}\sin\chi\, a^{3/2} \equiv S_\alpha$$

for r $$\frac{\partial r}{\partial r_0} = 1 + D_0\left(-\cos\chi\,\frac{\hat{S}}{r}\right) + C_0\left(-\frac{\hat{S}^2}{r}\right) + \alpha\hat{C} = \frac{r_0}{r} f$$

$$\frac{\partial r}{\partial D_0} = \hat{S} + D_0\left(-\cos\chi\,\frac{\hat{C}}{r}\right) + C_0\left(-\frac{\hat{S}\hat{C}}{r}\right) = \frac{1}{r} g$$

$$\frac{\partial r}{\partial \alpha} = D_0 S_\alpha + C_0 C_\alpha + r_0 \hat{C} \equiv r_\alpha$$

Finally, the required partials of f, g, $\dot{f}$, and $\dot{g}$, with respect to r_0, D_0, and α are:

for f $$\frac{\partial f}{\partial r_0} = \frac{\hat{C}}{r_0^2} + \frac{\hat{S}^2}{r r_0}$$

$$\frac{\partial f}{\partial D_0} = \frac{\hat{S}\hat{C}}{r r_0}$$

$$\frac{\partial f}{\partial \alpha} = -\frac{1}{r_0} C_\alpha$$

for g $$\frac{\partial g}{\partial r_0} = \frac{\hat{S}\hat{C}}{r}$$

$$\frac{\partial g}{\partial D_0} = \frac{\hat{C}^2}{r}$$

$$\frac{\partial g}{\partial \alpha} = \frac{1}{2}\alpha\left[\tau - 3\hat{U}\right] + \frac{\hat{C}}{r}\chi_\alpha$$

for $\dot{f}$ $$\frac{\partial \dot{f}}{\partial r_0} = \frac{\hat{S}}{r^3 r_0^2}\left[r^2 + r_0\left(r\cos\chi + r_0 f\right)\right]$$

$$\frac{\partial \dot{f}}{\partial D_0} = \frac{1}{r^3 r_0}\left[\hat{S} g + \cos\chi\, r\hat{C}\right]$$

$$\frac{\partial \dot{f}}{\partial \alpha} = \frac{1}{r r_0} S_\alpha + \frac{\hat{S}}{r^2 r_0} r_\alpha$$

for $\dot{g}$ $$\frac{\partial \dot{g}}{\partial r_0} = \frac{1}{r^3}\left[r\hat{S}^2 + \hat{C} r_0 f\right]$$

$$\frac{\partial \dot{g}}{\partial D_0} = \frac{\hat{C}}{r^3}\left[r\hat{S} + g\right]$$

$$\frac{\partial \dot{g}}{\partial \alpha} = -\frac{r C_\alpha + \hat{C} r_\alpha}{r^2}$$

The only remaining steps at this point are thus to provide the derivatives of the intermediate set of parameters with respect to the components of $\vec{r}_0$, and $\vec{s}_0$, to construct the derivatives of f, g, $\dot{f}$ and $\dot{g}$ with respect to $\vec{r}_0$ and $\vec{s}_0$, and to relate the results to the parameters $\vec{r}$ and $\vec{v}$. The first step is accomplished by referring to the definitions of r_0, D_0, and α.

$$r_0^2 = \sum_{i=1}^{3} x_i^2 \; ; \qquad s_0^2 = \sum_{i=1}^{3} s_i^2$$

$$\frac{\partial r_0}{\partial x_i} = \frac{x_i}{r_0} \; ; \qquad \frac{\partial r_0}{\partial s_i} = 0$$

$$D_0 = \vec{r}_0 \cdot \vec{s}_0$$

$$\frac{\partial D_0}{\partial x_i} = s_i \; ; \qquad \frac{\partial D_0}{\partial s_i} = x_i$$

$$\alpha = \vec{s} \cdot \vec{s} - \frac{2}{r_0}$$

$$\frac{\partial \alpha}{\partial x_i} = \frac{2 x_i}{r_0^3} \; ; \qquad \frac{\partial \alpha}{\partial s_i} = s_i$$

The second step is accomplished through the medium of the chain rule, i.e.,

$$\frac{\partial f}{\partial x_i} = \frac{\partial f}{\partial r_0} \frac{\partial r_0}{\partial x_i} + \frac{\partial f}{\partial D_0} \frac{\partial D_0}{\partial x_i} + \frac{\partial f}{\partial \alpha} \frac{\partial \alpha}{\partial x_i}$$

etc.

and the third and final step is employed to remove the normalization factor applied to the components of the velocity. Since

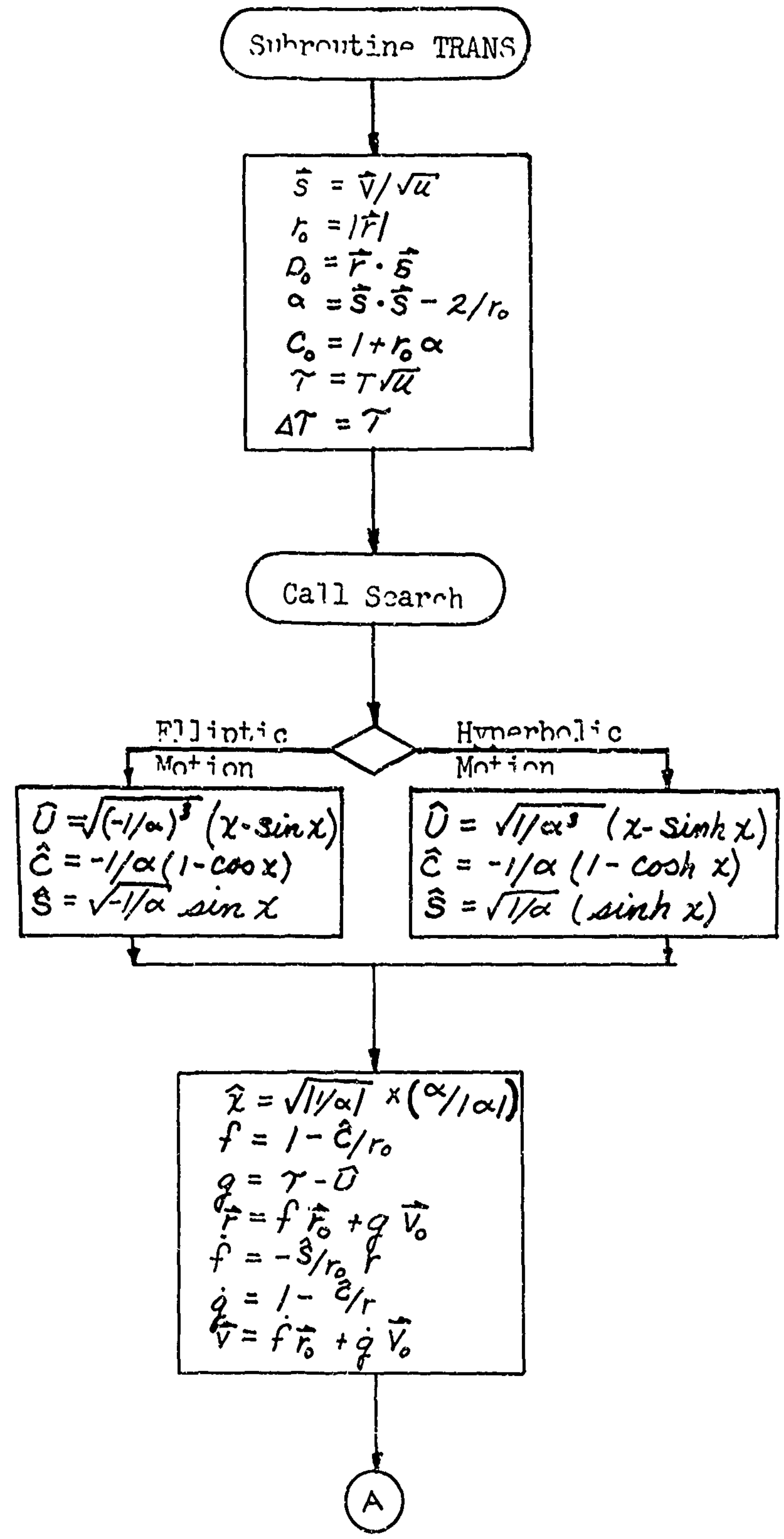
$$\vec{s} = \frac{\vec{V}}{\sqrt{\mu}}$$

$$d\vec{s} = \frac{1}{\sqrt{\mu}} \, d\vec{V} \quad ,$$

the desired matrix is

$$\begin{Bmatrix} d\vec{r} \\ d\vec{v} \end{Bmatrix} \quad \left[\begin{array}{c|c} \dfrac{\partial \vec{r}}{\partial \vec{r}_0} & \dfrac{1}{\sqrt{\mu}} \dfrac{\partial \vec{r}}{\partial \vec{s}_0} \\ \hline \sqrt{\mu} \, \dfrac{\partial \vec{s}}{\partial \vec{r}_0} & \dfrac{\partial \vec{s}}{\partial \vec{s}_0} \end{array} \right] \quad \begin{Bmatrix} d\vec{r}_0 \\ d\vec{v}_0 \end{Bmatrix}$$

Computational Logic:



A

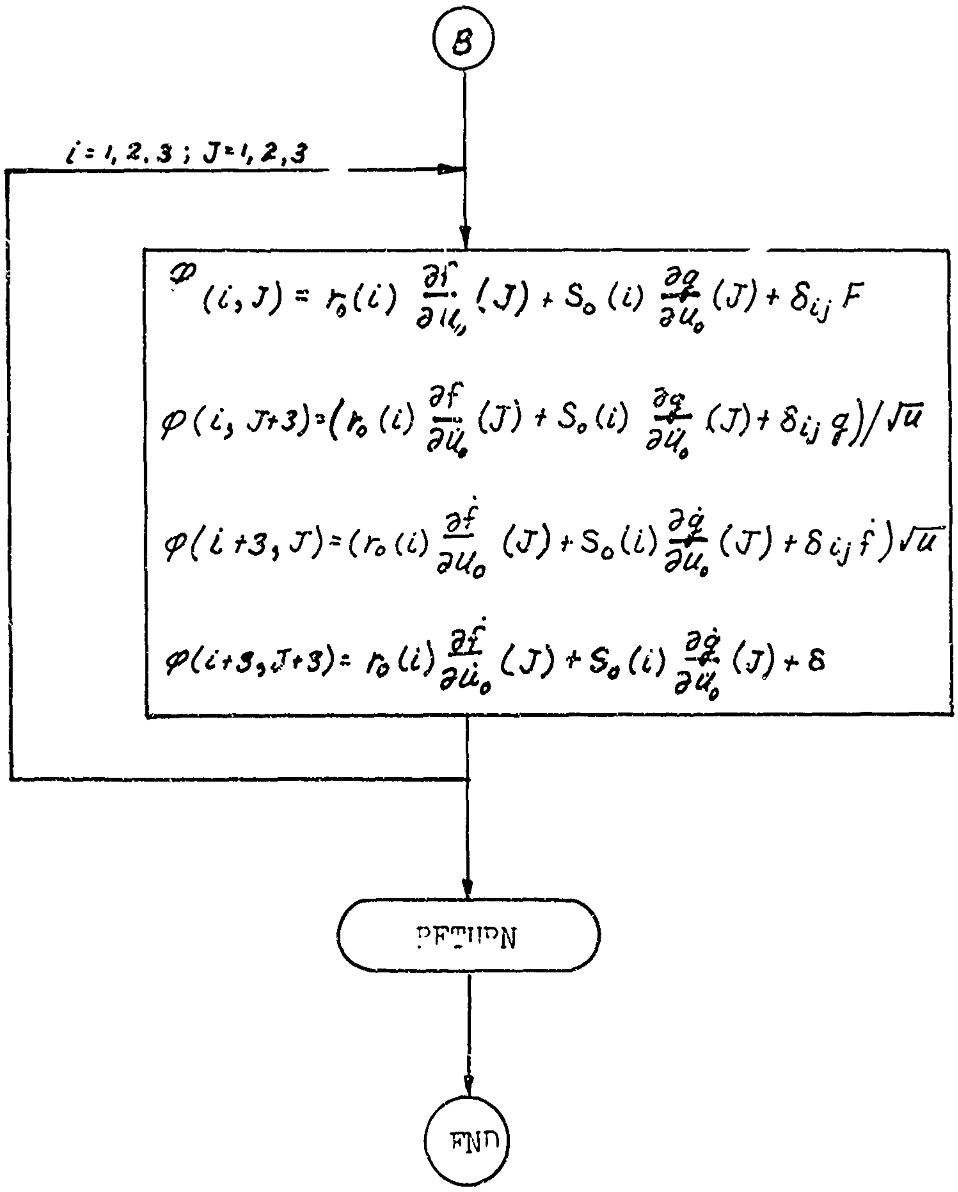
$$U^* = -\frac{1}{\alpha}\left(\hat{x}^3/6 - \hat{U}\right)$$
$$C^* = -\frac{1}{\alpha}\left(\hat{x}^2/2 - \hat{C}\right)$$
$$U_\alpha = \frac{1}{2}\left(\hat{x} C^* - 3U^*\right)$$
$$C_\alpha = \frac{1}{2}\left(\hat{x}\hat{U} - 2C^*\right)$$
$$X_\alpha = r_0 \hat{U} + D_0 C_\alpha + C_0 U_\alpha$$
$$K = 1 + \alpha \hat{C}$$
$$S_\alpha = \hat{U} + \alpha U_\alpha$$
$$r_\alpha = r_0 \hat{C} + D_0 S_\alpha + C_0 C_\alpha$$

$$\frac{\partial f}{\partial r}, \quad \frac{\partial f}{\partial D}, \quad \frac{\partial f}{\partial \alpha}$$
$$\frac{\partial q}{\partial r}, \quad \frac{\partial \dot{q}}{\partial D}, \quad \frac{\partial q}{\partial \alpha}$$
$$\frac{\partial \dot{f}}{\partial r}, \quad \frac{\partial \dot{f}}{\partial D}, \quad \frac{\partial \dot{f}}{\partial \alpha}$$

For u = x, y, z

$$\frac{\partial r}{\partial u_0} = \frac{r u_0}{r_0} \; ; \qquad \frac{\partial r}{\partial \dot{u}_0} = 0$$
$$\frac{\partial D}{\partial u_0} = S u_0 \; ; \qquad \frac{\partial D}{\partial \dot{u}_0} = r_u$$
$$\frac{\partial \alpha}{\partial u_0} = 2\frac{r u_0}{r_0^3} \; ; \qquad \frac{\partial \alpha}{\partial \dot{u}_0} = 2 S u$$
$$\frac{\partial f}{\partial u_0}(i) = \frac{\partial f}{\partial r_0}\frac{\partial r_0}{\partial u} + \frac{\partial f}{\partial D_0}\frac{\partial D_0}{\partial u} + \frac{\partial f}{\partial \alpha}\frac{\partial \alpha}{\partial u}$$
$$\vdots$$

B


B
$i = 1, 2, 3 \; ; \; J = 1, 2, 3$
$\varphi(i, J) = r_0(i) \frac{\partial f}{\partial u_0}(J) + S_0(i) \frac{\partial g}{\partial u_0}(J) + \delta_{ij} F$
$\varphi(i, J+3) = \left(r_0(i) \frac{\partial f}{\partial \dot{u}_0}(J) + S_0(i) \frac{\partial g}{\partial \dot{u}_0}(J) + \delta_{ij} g \right) / \sqrt{u}$
$\varphi(i+3, J) = \left(r_0(i) \frac{\partial \dot{f}}{\partial u_0}(J) + S_0(i) \frac{\partial \dot{g}}{\partial u_0}(J) + \delta_{ij} \dot{f} \right) \sqrt{u}$
$\varphi(i+3, J+3) = r_0(i) \frac{\partial \dot{f}}{\partial \dot{u}_0}(J) + S_0(i) \frac{\partial \dot{g}}{\partial \dot{u}_0}(J) + \delta$
RETURN
END

```
****
   TRANX       -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      SUBROUTINE TRANS(R,V,T,PHI)                                         TRAN0020
C                                                                         TRAN0030
C     THIS ROUTINE EVALUATES THE STATE TRANSITION MATRIX FOR CONIC        TRAN0040
C             MOTION THROUGH AN ARBITRARY TIME T     . THE COORDINATE     TRAN0050
C             FRAME FOR THE TRANSITION MATRIX IS INERTIAL AND MUST BE     TRAN0060
C             THE SAME AS THAT PRESCRIBED FOR R AND V .                   TRAN0070
C     THE VARIABLES OF THIS ANALYSIS ARE THE SAME AS THOSE UTILIZED IN    TRAN0080
C             SUBROUTINE CONIC AND WERE RECOMMENDED BY DR. HERRICK .      TRAN0090
C     R  = INITIAL POSITION VECTOR                                        TRAN0100
C     V  = INITIAL VELOCITY VECTOR                                        TRAN0110
C     T = TIME FROM (R0,V0 )                                              TRAN0120
C     PHI= TRANSITION MATRIX FOR CONIC MOTION                             TRAN0130
C                                                                         TRAN0140
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  TRAN0150
C                                                                         TRAN0160
      DIMENSION  CON(1), SAT(1), SDA(1), WRK(1), STT(1)                    TRAN0170
                                                                          TRAN0180
C                                                                         TRAN0190
      COMMON DATA                                                         TRAN0200
C                                                                         TRAN0210
      EQUIVALENCE  (DATA(  1),CON), (DATA( 16),SAT), (DATA( 36),SDA)     TRAN0220
     2            ,(DATA(286),STT), (DATA(391),WRK)                       TRAN0230
C                                                                         TRAN0240
C                                                                         
      EQUIVALENCE  (CON(6),GM),(WRK(130),ELEM)                            TRAN0250
C                                                                         TRAN0260
      DIMENSION  R(3),V(3),PHI(6,6),S(3),RF(3),VF(3)                      TRAN0270
     1,DFDU(3),DGDU(3),DFDDU(3),DGDDU(3),DFDV(3),DFDDV(3)                 TRAN0280
     2,DGDV(3),DGDDV(3),ELEM(4)                                           TRAN0290
C                                                                         TRAN0300
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  TRAN0310
C                                                                         TRAN0320
C     COMPUTATION OF THE ELEMENTS                                         TRAN0330
      SQG = SQRT( GM )                                                    TRAN0340    2
      AA= 1.                                                              TRAN0350
      DO 1 I=1,3                                                          TRAN0360
    1 S(I) = V(I)/ SQG                                                    TRAN0370
```

```
      TRANX        -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      RO = AMAG( R )                                                TRAN0380    11
      DO= DOT(R,S )                                                 TRAN0390    12
      ALPHA = DOT(S,S) - 2./RO                                      TRAN0400    13
      CO= 1. + RO * ALPHA                                           TRAN0410
      ELEM(1) = RO                                                  TRAN0420
      ELEM(2) = DO                                                  TRAN0430
      ELEM(3) = -1./ALPHA                                           TRAN0440
      ELEM(4) = CO                                                  TRAN0450
      TAU = T*SQG                                                   TRAN0460
      DTAU = TAU                                                    TRAN0470
      X = 0.                                                        TRAN0480
      CALL SEARCH(TAU,DTAU,X)                                       TRAN0490    14
      IF( ALPHA) 10,20,20                                           TRAN0500
   10 UHAT = SQRT( -1./(ALPHA)**3) * (X-SIN(X))                     TRAN0510    18     19
      CHAT = ELEM(3) * (1.-COS (X) )                                TRAN0520    20
      SHAT = SQRT( -1./ ALPHA) * SIN(X)                             TRAN0530    21     22
      GO TO 30                                                      TRAN0540
   20 UHAT =-SQRT( +1./(ALPHA)**3) * (X-SINH(X))                    TRAN0550    25     26
      CHAT = ELEM(3) * (1.-COSH(X) )                                TRAN0560    27
      SHAT =-SQRT( +1./(ALPHA))*SINH(X)                             TRAN0570    28     29
   30 XHAT =-SQRT(ABS( 1./ALPHA )) * X * SIGN( AA,ALPHA )            TRAN0580
C                                                                   TRAN0590
C                                                                   TRAN0600
C     COMPUTATION OF R AND V AT T                                   TRAN0610    31
      F = 1. - CHAT / RO                                            TRAN0620
      G = TAU- UHAT                                                 TRAN0630
      DO 40 I =1,3                                                  TRAN0640
   40 RF(I) = F * R(I) + G * S(I)                                   TRAN0650
      RT = AMAG(RF)                                                 TRAN0660    41
      FDOT = -SHAT/(RO*RT)                                          TRAN0670
      GDOT = 1. - CHAT/RT                                           TRAN0680
      DO 50 I =1,3                                                  TRAN0690
   50 VF(I) = FDOT * R(I) + GDOT * S(I)                             TRAN0700
C                                                                   TRAN0710
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * TRAN0720
C                                                                   TRAN0730
C     AT THIS POINT THE ELEMENTS HAVE BEEN EVALUATED AND ALL INFORMATIONTRAN0740
```

```
C              NECESSARY TO EVALUATE THE TRANSITION MATRIX IS AVAILABLE.TRAN0750
               FIRST,HOWEVER, SEVERAL FUNCTIONS WHICH ARE USED WILL BE  TRAN0760
C              DEFINED TO SIMPLIFY THE EXPRESSIONS.                     TRAN0770
C                                                                       TRAN0780
C                                                                       TRAN0790
      USTAR = ((XHAT**3)/6. - UHAT ) * ELEM(3)                          TRAN0800
      CSTAR = ((XHAT**2)/2. - CHAT ) * ELEM(3)                          TRAN0810
      UALF  = .5* ( XHAT * CSTAR - 3. * USTAR )                         TRAN0820
      CALF  = .5* ( XHAT * UHAT  - 2. * CSTAR )                         TRAN0830
      XALF  =  RO*UHAT + DO*CALF + CO*UALF                              TRAN0840
      CONST =  1. + ALPHA * CHAT                                        TRAN0850
      SALF  =  UHAT + ALPHA * UALF                                      TRAN0860
      RALF  =  RO * CHAT + DO * SALF + CO *CALF                         TRAN0870
C                                                                       TRAN0880
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * TRAN0890
C                                                                       TRAN0900
      THE DERIVATIVES OF ( F,FDOT,G,GDOT ) WITH RESPECT TO THE SET      TRAN0910
C              OF PARAMETERS (RO,DO,ALPHA )                             TRAN0920
C                                                                       TRAN0930
      DFDR = (CHAT/RO + SHAT*SHAT/RT ) / RO                             TRAN0940
      DFDD = (SHAT * CHAT )/( RO * RT )                                 TRAN0950
      DFDA = (SHAT * XALF - RT * CALF )/(RO * RT )                      TRAN0960
      DGDR =  DFDD * RO                                                 TRAN0970
      DGDD =  CHAT * CHAT / RT                                          TRAN0980
      DGDA = (CHAT * XALF - RT * UALF ) / RT                            TRAN0990
C                                                                       TRAN1000
C                                                                       TRAN1010
      DFDDR = SHAT/(RT**3 *RO**2) *( RT**2 +RO*(RT*CONST+RO*F ))        TRAN1020
      DFDDD =(SHAT*G + RT*CONST* CHAT ) / ( RT**3 * RO )                TRAN1030
      DFDDA =(RO *F *XALF + RT*( SHAT* RALF - RT *SALF ))/ ( RT**3 * RO)TRAN1040
      DGDDR =(RO *F *CHAT + RT*( SHAT**2 )) / ( RT **3 )                TRAN1050
      DGDDD = CHAT *( G + RT * SHAT ) / ( RT ** 3 )                     TRAN1060
      DGDDA =( G * XALF + RT *( CHAT *RALF - RT * SALF ) ) / (RT **3 )  TRAN1070
                                                                        TRAN1080
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * TRAN1090
C                                                                       TRAN1100
C     THE DERIVATIVES CAN NOW BE COMBINED TO COMPUTE THE VARIATIONAL    TRAN1110
```

302

```
C             COEFFICIENTS .                                                TRAN1120
C                                                                           TRAN1130
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * TRAN1140
C                                                                           TRAN1150
C     THE DERIVATIVES OF THE SET (RO,DO,ALPHA ) WITH RESPECT TO THE         TRAN1160
C             SET ( X,Y,Z,XDOT,YDOT,ZDOT )                                  TRAN1170
C                                                                           TRAN1180
      DO 60 I=1,3                                                           TRAN1190
      DRDX = R(I)/ RO                                                       TRAN1200
      DDDX = S(I)                                                           TRAN1210
      DADX = 2.*R(I)/ RO**3                                                 TRAN1220
      DRDV = 0.                                                             TRAN1230
      DDDV = R(I)                                                           TRAN1240
      DADV = 2.* S(I)                                                       TRAN1250
C                                                                           TRAN1260
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * TRAN1270
C                                                                           TRAN1280
C     THE DERIVATIVES OF THE SET (F,FDOT,G,GDOT ) WITH RESPECT TO THE       TRAN1290
C             SET ( X,Y,Z,XDOT,YDOT,ZDOT,)                                  TRAN1300
      DFDU(I) = DFDR * DRDX  + DFDD * DDDX  + DFDA * DADX                   TRAN1310
      DGDU(I) = DGDR * DRDX  + DGDD * DDDX  + DGDA * DADX                   TRAN1320
      DFDDU(I)= DFDDR* DRDX  + DFDDD* DDDX  + DFDDA* DADX                   TRAN1330
      DGDDU(I)= DGDDR* DRDX  + DGDDD* DDDX  + DGDDA* DADX                   TRAN1340
C                                                                           TRAN1350
C                                                                           TRAN1360
      DFDV(I) =              + DFDD * DDDV  + DFDA * DADV                   TRAN1370
      DGDV(I) =              + DGDD * DDDV  + DGDA * DADV                   TRAN1380
      DFDDV(I)=              + DFDDD* DDDV  + DFDDA* DADV                   TRAN1390
      DGDDV(I)=              + DGDDD* DDDV  + DGDDA* DADV                   TRAN1400
   60 CONTINUE                                                              TRAN1410
C                                                                           TRAN1420
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * TRAN1430
C                                                                           TRAN1440
C     THE TRANSITION MATRIX CAN NOW BE COMPUTED                             TRAN1450
      DO 70 I=1,3                                                           TRAN1460
      DO 70 J=1,3                                                           TRAN1470
      K= J+3                                                                TRAN1480
```

```
      TRANX       -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      L= I+3                                                      TRAN1490
      PHI(I,J) = R(I) * DFDU(J) + S(I) * DGDU(J) +DERAQ(I,J) * F   TRAN1500   77
      PHI(I,K) =(R(I) * DFDV(J) + S(I) * DGDV(J) +DERAQ(I,J) * G)/SQG  TRAN1510   83
      PHI(L,J) =(R(I) * DFDDU(J)+ S(I) * DGDDU(J)+DERAQ(I,J)*FDOT)*SQG  TRAN1520   89
      PHI(L,K) = R(I) * DFDDV(J)+ S(I) * DGDDV(J) +DERAQ(I,J)*GDOT  TRAN1530   95
   70 CONTINUE                                                    TRAN1540
C                                                                 TRAN1550
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * TRAN1560
C                                                                 TRAN1570
      RETURN                                                      TRAN1580
      END                                                         TRAN1590
```

STORAGE MAP

SUBROUTINE TRANS

COMMON VARIABLES

COMMON BLOCK // ORIGIN 00001 LENGTH 01013

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DATA	00000	R	DATA	00000	R	CON	00000	R
SAT	00017	R	SDA	00043	R	STT	00435	R
WRK	00606	R	GM	00005	R	ELEM	01007	R

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
S	01014	R	RF	01017	R	VF	01022	R
DFDU	01025	R	DGDU	01030	R	DFDDU	01033	R
DGDDU	01036	R	DFDV	01041	R	DFDDV	01044	R
DGDV	01047	R	DGDDV	01052	R			

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
SQG	01055	R	AA	01056	R	I	01057	I
RO	01060	R	DO	01061	R	ALPHA	01062	R
CO	01063	R	TAU	01064	R	DTAU	01065	R
X	01066	R	UHAT	01067	R	CHAT	01070	R
SHAT	01071	R	XHAT	01072	R	F	01073	R
G	01074	R	RT	01075	R	FDOT	01076	R
GDOT	01077	R	USTAR	01100	R	CSTAR	01101	R
UALF	01102	R	CALF	01103	R	XALF	01104	R
CONST	01105	R	SALF	01106	R	RALF	01107	R
DFDR	01110	R	DFDD	01111	R	DFDA	01112	R
DGDR	01113	R	DGDD	01114	R	DGDA	01115	R
DFDDR	01116	R	DFDDD	01117	R	DFDDA	01120	R

STORAGE MAP

DGDDR	01121	R	DGDDD	01122	R	DGDDA	01123	R
DRDX	01124	R	DDDX	01125	R	DADX	01126	R
DRDV	01127	R	DDDV	01130	R	DADV	01131	R
J	01132	I	K	01133	I	L	01134	I

ENTRY POINTS

TRANS	SECTION	4

SUBROUTINES CALLED

SQRT	SECTION	5	AMAG	SECTION	6	DOT	SECTION	7
SEARCH	SECTION	8	SIN	SECTION	9	COS	SECTION	10
SINH	SECTION	11	COSH	SECTION	12	DERAQ	SECTION	13
SYSLOC	SECTION	14						

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1	6A	01165	10	17A	01257	20	24A	01342
30	30A	01422	40	35A	01461	50	45A	01516
60	67A	02237	70	101A	02417			

DECK LENGTH IN OCTAL IS 01465.

2.4.1.2.2 Subroutine INVAO

Purpose: to produce the analytic inverse of the 6 X 6 matrix of partial derivatives referred to as the state transition matrix (i.e., the matrix $\left[\frac{\partial X}{\partial X_0}\right]$ where all vectors are expressed in cartesian coordinates)

Deck Name: INVA

Calling Sequence: Call INVAO (AO, AOI)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	AO	$\varphi(t,t_0)$	6 X 6	ARG	the matrix of partial derivatives of the state vector ($\vec{x}$) at "t" with respect to the state vector at "t_o"
O	AOI	$\varphi^{-1}(t,t_0)$ or $\varphi(t_0,t)$	6 X 6	ARG	the analytically inverted matrix

Subroutine required: None

Functions required: None

Approximate Deck Length: 160 (octal)

Formulation:

Consider a linear system (expressed in cartesian coordinates) described by the following equation

$$\dot{\vec{X}} = A(t)\,\vec{X}(t) + B(t)\,\vec{F}(t) \qquad (1)$$

where $\vec{X}(t)$ is an even-ordered state vector composed of a set of output variables and their derivatives

$\vec{F}(t)$ is a forcing function (= 0 for the analysis to follow)
$A(t)$ is a coefficient array for the system composed of square, symmetric, even-ordered subarrays of the following form

$$A(t) = \left[\begin{array}{c|c} 0 & A_{12} \\ \hline A_{21} & 0 \end{array}\right] \tag{2}$$

$$A_{12} = I$$

$$A_{21} = \frac{\partial \vec{g}}{\partial \vec{r}} = \frac{\mu}{r^3}\left[I - \frac{3\vec{r}\,\vec{r}^{\,T}}{r^2}\right]$$

$B(t)$ is an array relating the sensitivity of the system to the forcing function
and the solution for $\vec{F} = 0$

$$\vec{X}(t) = \varphi(t, t_0)\ \vec{X}(t_0) \equiv \left[\begin{array}{c|c} \varphi_{11} & \varphi_{12} \\ \hline \varphi_{21} & \varphi_{22} \end{array}\right] \vec{X}(t_0) \tag{3}$$

If equations 2 and 3 are substituted back into equation 1, the following identity results

$$\begin{bmatrix} \dot{\varphi}_{11} & \dot{\varphi}_{12} \\ \dot{\varphi}_{21} & \dot{\varphi}_{22} \end{bmatrix} = \begin{bmatrix} 0 & I \\ \frac{\partial \vec{g}}{\partial \vec{r}} & 0 \end{bmatrix} \begin{bmatrix} \varphi_{11} & \varphi_{12} \\ \varphi_{21} & \varphi_{22} \end{bmatrix} \tag{4}$$

and equation 4 yields upon expansion

$$\dot{\varphi}_{11} = \varphi_{21} \tag{5a}$$

$$\dot{\varphi}_{12} = \varphi_{22} \tag{5b}$$

$$\dot{\varphi}_{21} = \frac{\partial \vec{g}}{\partial r}\,\varphi_{11} \tag{5c}$$

$$\dot{\varphi}_{22} = \frac{\partial \vec{g}}{\partial r}\,\varphi_{12} \tag{5d}$$

Equations 5 may now be operated on to produce an equivalent set of differential equations. This operation is performed as follows:

$$\varphi_{21}^{T}\,\dot{\varphi}_{11} = \varphi_{21}^{T}\,\varphi_{21}$$

$$\dot{\varphi}_{21}^{T}\,\varphi_{11} = \varphi_{11}^{T}\,\frac{\partial \vec{g}}{\partial r}^{T}\,\varphi_{11} = \varphi_{11}^{T}\,\frac{\partial \vec{g}}{\partial \vec{r}}\,\varphi_{11}$$

$$\varphi_{11}^T \dot{\varphi}_{21} = \varphi_{11}^T \frac{\partial \vec{g}}{\partial \vec{r}} \varphi_{11}$$

$$\dot{\varphi}_{11}^T \varphi_{21} = \varphi_{21}^T \varphi_{21}$$

or

$$\frac{d}{dt}\left(\varphi_{21}^T \varphi_{11} - \varphi_{11}^T \varphi_{21}\right) = 0$$

$$\varphi_{21}^T \varphi_{11} - \varphi_{11}^T \varphi_{21} = C_1 \tag{6a}$$

similarly

$$\varphi_{22}^T \varphi_{11} - \varphi_{12}^T \varphi_{21} = C_2 \tag{6b}$$

$$\varphi_{22}^T \varphi_{12} - \varphi_{12}^T \varphi_{22} = C_3 \tag{6c}$$

$$\varphi_{11}^T \varphi_{22} - \varphi_{21}^T \varphi_{12} = C_4 \tag{6d}$$

Finally, the results of the integration can be restated in matrix notation as

$$\begin{bmatrix} \varphi_{22}^T & -\varphi_{12}^T \\ -\varphi_{22}^T & \varphi_{11}^T \end{bmatrix} \begin{bmatrix} \varphi_{11} & \varphi_{12} \\ \varphi_{21} & \varphi_{22} \end{bmatrix} = \begin{bmatrix} C_2 & C_3 \\ -C_1 & C_4 \end{bmatrix} \tag{7}$$

and the constant arrays resulting from these integrations may now be evaluated by substituting the initial conditions

$$\varphi_{11}(0) = \varphi_{22}(0) = I$$

$$\varphi_{12}(0) = \varphi_{21}(0) = 0$$

This step produces

$$C_1 = 0$$

$$C_2 = I$$

$$C_3 = 0$$

$$C_4 = I$$

and reduces equation 7 to the identity matrix. But, the only matrix which can be utilized to reduce an arbitrary square matrix to I is its inverse. Thus

$$\varphi^{-1}(t, t_o) = \begin{bmatrix} \varphi_{22}^T & -\varphi_{12}^T \\ -\varphi_{21}^T & \varphi_{11}^T \end{bmatrix} \tag{8}$$

Equation 8 is important for general linear systems in that it provides an analytic means of constructing the inverse transition matrix directly from the elements of the known transition matrix by rearrangement of terms and the change of a few signs. It will be utilized in this program in conjunction with a correction to the basic conic transition matrix descirbed in STMAT.

In conclusion, it is noted that the true meaning of the terms A_{12} and A21 (equation 2) was never employed, and only symmetry is required. Thus, there is an immediate generalization providing that an arbitrary system can be described by equation 2. It is also noted that another approach to the derivation is suggested in Ref. 1, along with a discussion of several related problems.

References

1) Friedlander, A. L., "Inversion Property of the Fundamental Matrix in Trajectory Perturbation Problems." AIAA Journal, Vol. 1, No. 4, pages 971-973(April 1963)

```
      SUBROUTINE INVAC(AC,AOI)                                          INVA0020
C                                                                       INVA0030
C     THIS ROUTINE COMPUTES THE INVERSE OF THE STATE TRANSITION MATRIX  INVA0040
C             UTILIZING A THEORM VALID FOR THIS TYPE OF MATRIX.         INVA0050
C                                                                       INVA0060
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  INVA0070
C                                                                       INVA0080
      DIMENSION AO(6,6),AOI(6,6)                                        INVA0090
C                                                                       INVA0100
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  INVA0110
C                                                                       INVA0120
      DO 1  I=1,3                                                       INVA0130
      II = I + 3                                                        INVA0140
      DO 2 J=1,3                                                        INVA0150
      JJ = J + 3                                                        INVA0160
      AOI(I,J) = AO(JJ,II)                                              INVA0170
      AOI(II,J) = -AO(JJ,I)                                             INVA0180
      AOI(II,JJ) = AO(J,I)                                              INVA0190
      AOI(I,JJ) = -AO(J,II)                                             INVA0200
    2 CONTINUE                                                          INVA0210
    1 CONTINUE                                                          INVA0220
      RETURN                                                            INVA0230
      END                                                               INVA0240
```

STØRAGE MAP

SUBRØUTINE INVAØ

UNDIMENSIØNED PRØGRAM VARIABLES

SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE
I	00001	I	II	00002	I	J	00003	I
JJ	00004	I						

ENTRY PØINTS

INVAØ SECTIØN 3

SUBRØUTINES CALLED

SYSLØC SECTIØN 4

EFN IFN CØRRESPØNDENCE

EFN	IFN	LØCATIØN	EFN	IFN	LØCATIØN	EFN	IFN	LØCATIØN
1	19A	00113	2	17A	00110			

DECK LENGTH IN ØCTAL IS 00157.

2.4.1.3 Subroutine MEASUR

Purpose: To construct the matrix of partial derivatives of the observables with respect to the state vector (i.e., $M = \frac{\partial \vec{Y}}{\partial \vec{x}}$) where the observables may be

1) range
2) range rate
3) azimuth and elevation
4) range and range rate
5) range, aximuth and elevation
6) range rate, aximuth and elevation

Deck Name: OBSN

Calling Sequence: CALL MEASUR (M)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Description
I	ITYPE	-	1	WRK(64)	Fixed point number which identifies the type of data being processed and hence, the type of matrix desired.
I	RVEC	$\vec{r}$	3	WRK(111)	Radius vector in frame of date
I	VVEC	$\vec{v}$	3	WRK(114)	Velocity vector in frame of date
I	RTRAK	$\vec{r}_t$	3	WRK(95)	Radius vector of tracking station in frame of date
I	X (Y,Z)	U (E,N)	3 (3,3)	WRK(108) WRK(124) WRK(127)	Up (east, north) unit vectors at the tracking station
O	M	-	1	ARG	Fixed point integer which identifies the number of pieces of data in the observation vector (1, 2 or 3)

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
O	OBST	H^m	[illegible] X 3	STT (37)	array of partial derivatives of the observations with respect to the state vector
I	OMEGA	ω	1	CON (7)	spin rate of the earth

Subroutines required:	CROSS	(cross product)
Functions required:	AMAG	(vector magnitude)
	DOT	(dot product)
	SQRT	(square root)
Approximate Deck Length:	350	(decimal)

Formulation:

Partials for range, range rate, azimuth and elevation with respect to the state vector can be obtained from the following equations:

$$R = (\vec{r}_r \cdot \vec{r}_r)^{1/2}$$

$$\dot{R} = (\hat{r}_r \cdot \vec{V}_r)$$

$$A_z = \tan^{-1}\left(\frac{\vec{r}_r \cdot \vec{E}}{\vec{r}_r \cdot \vec{N}}\right)$$

$$EL = \sin^{-1}(\hat{r}_r \cdot \hat{U})$$

$$\vec{X} = \vec{r} - \vec{r}_n$$

$$\vec{r}_r = \vec{r} - \vec{r}_T$$

$$\hat{r}_r = \vec{r}_r / R$$

$$\vec{V}_r = \vec{V} - \vec{V}_1$$

where:

R = range

$\dot{R}$ = range rate

A_z = azimuth

EL = elevation

$\hat{U},\hat{E},\hat{N}$ = up, east, north unit vectors

$\vec{r}$ = vehicles position in equatorial frame of date

$\vec{r}_n$ = nominal position on reference orbit

$\vec{r}_T$ = tracking stations position vector

when it is noted that for the purpose of differentiation, the nominal position vector and the tracking station position vector are constant, i.e.,

$$d\vec{x} = d\vec{r}_r \quad .$$

This set of operations has been performed, and the results of the analysis are presented below:

1) Partials of range

$$R^2 = X_r^2 + Y_r^2 + Z_r^2$$

$$\frac{\partial R}{\partial \vec{X}} = \left(\frac{\partial R}{\partial \vec{r}}, \frac{\partial R}{\partial \vec{V}}\right) = \left(\frac{X_r}{R}, \frac{X_r}{R}, \frac{Z_r}{R}, 0, 0, 0\right)$$

2) Partials of range-rate

$$\dot{R} = \frac{X_r}{R}\dot{X}_r + \frac{Y_r}{R}\dot{Y}_r + \frac{Z_r}{R}\dot{Z}_r$$

$$\frac{\partial \dot{R}}{\partial \vec{X}} = \left(\frac{\partial \dot{R}}{\partial \vec{r}}, \frac{\partial \dot{R}}{\partial \vec{V}}\right) = \Big(\frac{\dot{X}_r}{R} - \frac{\dot{R}X_r}{R^2}, \frac{\dot{Y}_r}{R} - \frac{\dot{R}\dot{X}_r}{R^2},$$

$$\frac{\dot{Z}_r}{R} - \frac{\dot{R}Z_r}{R^2}, \frac{X_r}{R}, \frac{Y_r}{R}, \frac{Z_r}{R}\Big)$$

3) Partials of Azimuth

$$S^2 = (\vec{r}_r \cdot \hat{E})^2 + (\vec{r}_r \cdot \hat{N})^2$$

$$\frac{\partial A_z}{\partial \vec{X}} = \left(\frac{\partial A_z}{\partial \vec{r}}, \frac{\partial A_z}{\partial \vec{V}}\right) = \Big[E_1 \frac{(\hat{N} \cdot \vec{r}_r)}{S^2} - N_1 \frac{(\hat{E} \cdot \vec{r}_r)}{S^2}$$

$$E_2 \left(\frac{\hat{N}\cdot\vec{r}_r}{S^2}\right) - N_2 \left(\frac{\hat{E}\cdot\vec{r}_r}{S^2}\right),$$

$$E_3 \left(\frac{\hat{N}\cdot\vec{r}_r}{S^2}\right) - N_3 \left(\frac{\hat{E}\cdot\vec{r}_r}{S^2}\right),$$

$$0, 0, 0 \Big]$$

4) Partials of Elevation

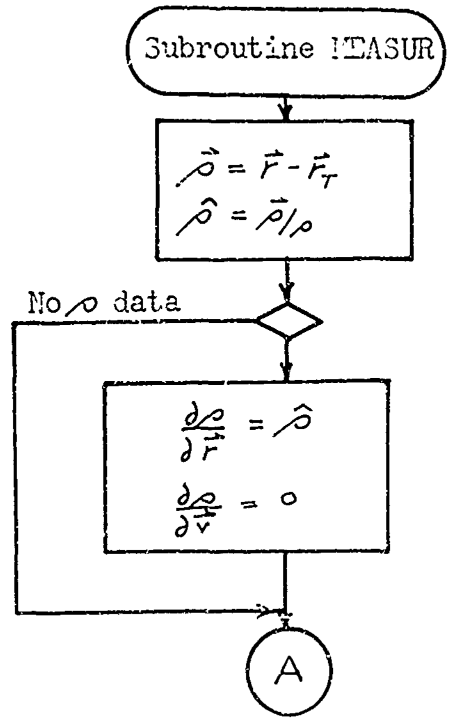
$$\frac{\partial EL}{\partial \bar{X}} = \left(\frac{\partial EL}{\partial \vec{r}}, \frac{\partial EL}{\partial \vec{v}}\right) = \Big[\frac{U_1}{S} - X_r \left(\frac{\hat{U}\cdot\vec{r}_r}{R^2 S}\right),$$

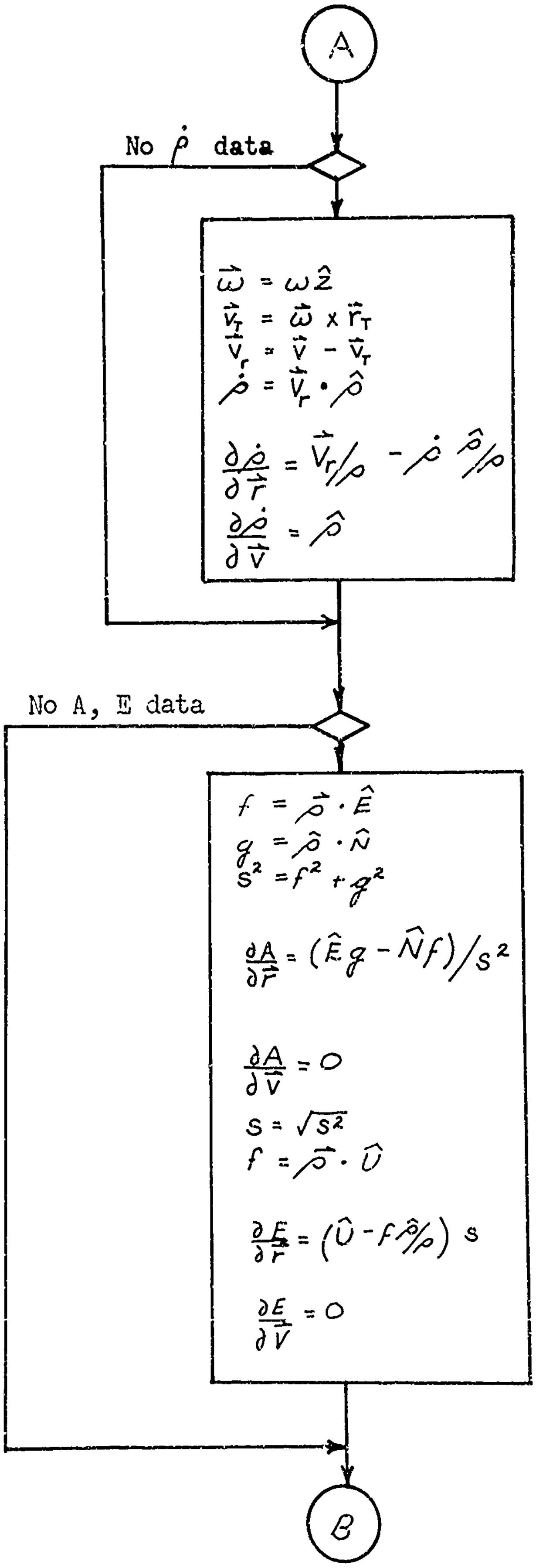
$$\frac{U_2}{S} - Y_r \left(\frac{\hat{U}\cdot\vec{r}_r}{R^2 S}\right),$$

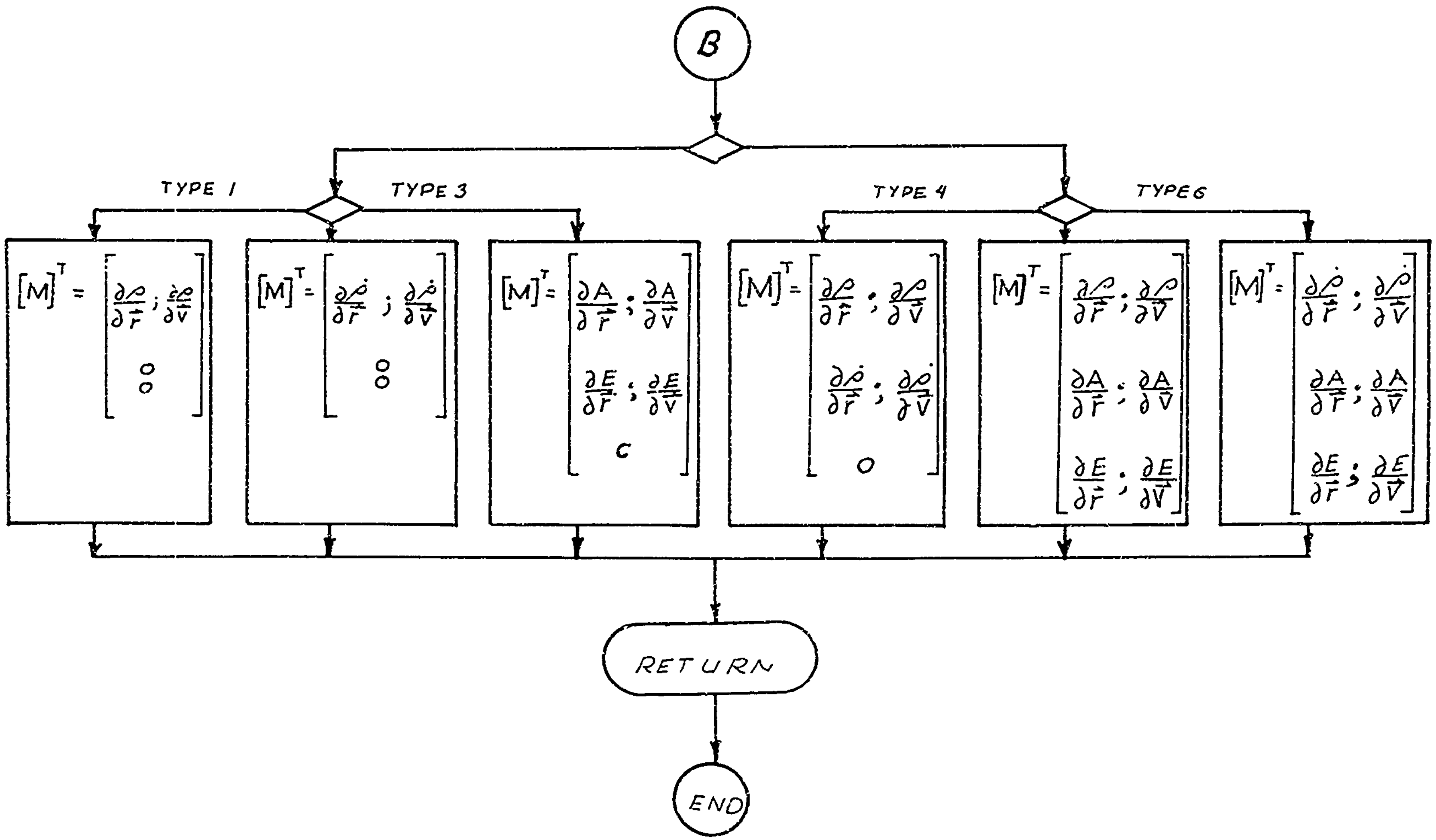
$$\frac{U_3}{S} - Z_r \left(\frac{\hat{U}\cdot\vec{r}_r}{R^2 S}\right),$$

$$0, 0, 0 \Big]$$

Computational Logic:




A
No $\dot{\rho}$ data
$\vec{\omega} = \omega \hat{z}$
$\vec{V}_T = \vec{\omega} \times \vec{r}_T$
$\vec{V}_r = \vec{V} - \vec{V}_T$
$\dot{\rho} = \vec{V}_r \cdot \hat{\rho}$
$\frac{\partial \dot{\rho}}{\partial \vec{r}} = \vec{V}_r/\rho - \dot{\rho}\,\hat{\rho}/\rho$
$\frac{\partial \dot{\rho}}{\partial \vec{V}} = \hat{\rho}$
No A, E data
$f = \vec{\rho} \cdot \hat{E}$
$g = \hat{\rho} \cdot \hat{N}$
$s^2 = f^2 + g^2$
$\frac{\partial A}{\partial \vec{r}} = (\hat{E} g - \hat{N} f)/s^2$
$\frac{\partial A}{\partial \vec{V}} = 0$
$s = \sqrt{s^2}$
$f = \vec{\rho} \cdot \hat{U}$
$\frac{\partial E}{\partial \vec{r}} = (\hat{U} - f\hat{\rho}/\rho)\, s$
$\frac{\partial E}{\partial \vec{V}} = 0$
B


B
TYPE 1
TYPE 3
TYPE 4
TYPE 6
$[M]^T = \begin{bmatrix} \frac{\partial \rho}{\partial \vec{r}} ; \frac{\partial \rho}{\partial \vec{V}} \\ 0 \\ 0 \end{bmatrix}$
$[M]^T = \begin{bmatrix} \frac{\partial \dot{\rho}}{\partial \vec{r}} ; \frac{\partial \dot{\rho}}{\partial \vec{V}} \\ 0 \\ 0 \end{bmatrix}$
$[M]^T = \begin{bmatrix} \frac{\partial A}{\partial \vec{r}} ; \frac{\partial A}{\partial \vec{V}} \\ \frac{\partial E}{\partial \vec{r}} ; \frac{\partial E}{\partial \vec{V}} \\ 0 \end{bmatrix}$
$[M]^T = \begin{bmatrix} \frac{\partial \rho}{\partial \vec{r}} ; \frac{\partial \rho}{\partial \vec{V}} \\ \frac{\partial \dot{\rho}}{\partial \vec{r}} ; \frac{\partial \dot{\rho}}{\partial \vec{V}} \\ 0 \end{bmatrix}$
$[M]^T = \begin{bmatrix} \frac{\partial \rho}{\partial \vec{r}} ; \frac{\partial \rho}{\partial \vec{V}} \\ \frac{\partial A}{\partial \vec{r}} ; \frac{\partial A}{\partial \vec{V}} \\ \frac{\partial E}{\partial \vec{r}} ; \frac{\partial E}{\partial \vec{V}} \end{bmatrix}$
$[M]^T = \begin{bmatrix} \frac{\partial \dot{\rho}}{\partial \vec{r}} ; \frac{\partial \dot{\rho}}{\partial \vec{V}} \\ \frac{\partial A}{\partial \vec{r}} ; \frac{\partial A}{\partial \vec{V}} \\ \frac{\partial E}{\partial \vec{r}} ; \frac{\partial E}{\partial \vec{V}} \end{bmatrix}$
RETURN
END

318

```
      SUBROUTINE MEASUR(M)                                              OBSN0020
C                                                                       OBSN0030
C     THIS ROUTINE EVALUATES THE MATRIX OF THE PARTIALS OF THE          OBSN0040
C              OBSERVATIONS WITH RESPECT TO THE STATE .THIS MATRIX      OBSN0050
C              WILL BE UTILIZED IN SUBROUTINE KALMAN DURING THE         OBSN0060
C              REDUCTION OF OBSERVED DATA.                              OBSN0070
C     ITYPE DENOTES WHICH OF THE SEVERAL TYPES OF DATA IS CONSIDERED    OBSN0080
C              1 = RANGE ONLY                                           OBSN0090
C              2 = RANGE RATE ONLY                                      OBSN0100
C              3 = AZIMUTH AND ELEVATION                                OBSN0110
C              4 = RANGE AND RANGE-RATE                                 OBSN0120
C              5 = RANGE,AZIMUTH AND ELEVATION                          OBSN0130
C              6 = RANGE-RATE,AZIMUTH AND ELEVATION                     OBSN0140
C     RVEC IS THE VECTOR POSITION IN THE FRAME OF DATE                  OBSN0150
C     VVEC IS THE VECTOR VELOCITY IN THE FRAME OF DATE                  OBSN0160
C     RTRAK IS THE POSITION VECTOR OF THE TRACKING STATION IN THE       OBSN0170
C              TRUE EQUATOR OF DATE FRAME                               OBSN0180
C     X,Y,Z ARE ZENITH,EAST,AND NORTH UNIT VECTORS AT THE STATION       OBSN0190
C     OBSN IS THE M*6 MATRIX OF OBSERVATION PARTIALS ( M IS THE         OBSN0200
C              DIMENSIONALITY OF THE OBSERVATION ) . THE ORDER OF THE   OBSN0210
C              STATE VECTOR IS ASSUMED ( X,Y,Z,XDOT,YDOT,ZDOT ).        OBSN0220
C              THE ORDER OF OBVEC IS RHO,RHODOT,AZIMUTH,ELEVATION       OBSN0230
C                                                                       OBSN0240
C                                                                       OBSN0250
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * OBSN0260
C                                                                       OBSN0270
      DIMENSION  CON(1), SAT(1), SDA(1), WRK(1), STT(1)                 OBSN0280
C                                                                       OBSN0290
      COMMON DATA                                                       OBSN0300
C                                                                       OBSN0310
      EQUIVALENCE  (DATA(  1),CON), (DATA( 16),SAT), (DATA( 36),SDA)    OBSN0320
     2            ,(DATA(286),STT), (DATA(391),WRK)                     OBSN0330
C                                                                       OBSN0340
      DIMENSION  RVEC(3)    ,VVEC(3)    ,RTRAK(3)   ,X(3)      ,Y(3)    OBSN0350
     2,Z(3)      ,OBST(6,3) ,RHO(3)     ,UNIT(3)    ,HR(6)     ,XN(3)   OBSN0360
     3,VTRAK(3) ,VREL(3)    ,HV(6)      ,HA(6)      ,HE(6)              OBSN0370
```

```
C                                                                     OBSN0380
      EQUIVALENCE           (CON(  7),OMEGA )                         OBSN0390
      EQUIVALENCE           (WRK( 64),ITYPE ), (WRK( 95),RTRAK )      OBSN0400
     2  ,(WRK(108),X      ), (WRK(124),Y     ), (WRK(127),Z     )     OBSN0410
     3  ,(WRK(111),RVEC   ), (WRK(114),VVEC  )                        OBSN0420
      EQUIVALENCE           (STT( 37),OBST  )                         OBSN0430
C                                                                     OBSN0440
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * OBSN0450
C                                                                     OBSN0460
C     THE FIRST STEP IS THE COMPUTE SOME OF THE PARTIALS COMMON TO THE OBSN0470
C             VARIOUS TYPES OF MEASUREMENTS . FIRST CONSTRUCT VECTOR  OBSN0480
C             TO SATELLITE FROM STATION                               OBSN0490
      DO 1 I=1,3                                                      OBSN0500
    1 RHO(I) = RVEC(I) - RTRAK(I)                                     OBSN0510
      RHOM = AMAG(RHO)                                                OBSN0520     11
      DO 2 I=1,3                                                      OBSN0530
    2 UNIT(I)=RHO(I)/ RHOM                                            OBSN0540
C                                                                     OBSN0550
C     RANGE DATA ARE INCLUDED                                         OBSN0560
      IF( ITYPE.EQ.2.OR.ITYPE.EQ.3.OR.ITYPE.EQ.6 ) GO TO 10           OBSN0570
      DO 3 I=1,3                                                      OBSN0580
      HR(I) = UNIT(I)                                                 OBSN0590
    3 HR(I+3) = 0.                                                    OBSN0600
C                                                                     OBSN0610
C     RANGE-RATE DATA ARE INCLUDED                                    OBSN0620
   10 IF( ITYPE.EQ.1.OR.ITYPE.EQ.3.OR.ITYPE.EQ.5 ) GO TO 20           OBSN0630
      XN(1) = 0.                                                      OBSN0640
      XN(2) = 0.                                                      OBSN0650
      XN(3) = OMEGA                                                   OBSN0660
      CALL CROSS( XN,RTRAK,VTRAK)                                     OBSN0670     35
      DO 11 I=1,3                                                     OBSN0680
   11 VREL(I) = VVEC(I) - VTRAK(I)                                    OBSN0690
      RHODOT = DOT(VREL,UNIT)                                         OBSN0700     46
      DO 12 I=1,3                                                     OBSN0710
      HV(I) = VREL(I)/RHOM - RHODOT* RHO(I)/ (RHOM * RHOM )           OBSN0720
   12 HV(I+3)= UNIT(I)                                                OBSN0730
C                                                                     OBSN0740
```

```
C      AZIMUTH AND ELEVATION ARE INCLUDED                             OBSN0750
   20 IF( ITYPE.EQ.1.OR.ITYPE.EQ.2.OR.ITYPE.EQ.4 ) GO TO 30           OBSN0760
      F = DOT(RHO,Y)                                                  OBSN0770    61
      G = DOT(RHO,Z)                                                  OBSN0780    62
      S2=  F*F + G*G                                                  OBSN0790
      DO 21 I=1,3                                                     OBSN0800
      HA(I) =(Y(I)*G - F*Z(I))/ S2                                    OBSN0810
   21 HA(I+3)= 0.                                                     OBSN0820
      S =SQRT(S2)                                                     OBSN0830    73
      F = DOT(X,RHO )                                                 OBSN0840    74
      DO 22 I=1,3                                                     OBSN0850
      HE(I)= ( X(I) -F*RHO(I)/( RHOM * RHOM))/ S                      OBSN0860
   22 HE(I+3)= 0.                                                     OBSN0870
C                                                                     OBSN0880
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * OBSN0890
C                                                                     OBSN0900
C     AT THIS POINT ALL OF THE NECESSARY PARTIALS ARE COMPUTED AND THE  OBSN0910
C              COMPLETE ARRAY CAN BE ASSEMBLED FOR ANY OF THE DATA      OBSN0920
C              TYPES BEING CONSIDERED . THIS OPERATION WILL BE MODIFIED OBSN0930
C              FOR STORAGE IN COMMON BY FINDING THE TRANSPOSE OF OBS.   OBSN0940
C              BECAUSE OF THIS OPERATION AN INVERSE SERIES OF           OBSN0950
C              STEPS MUST BE TAKEN IN KLMAN TO UNSCRAMBLE THE DATA      OBSN0960
C                                                                     OBSN0970
   30 DO 40 J=1,6                                                     OBSN0980
      GO TO (42,44,46,48,50,52 ),ITYPE                                OBSN0990
   42 M=1                                                             OBSN1000
      OBST(J,1) = HR(J)                                               OBSN1010
      OBST(J,2) = 0.                                                  OBSN1020
      OBST(J,3) = 0.                                                  OBSN1030
      GO TO 40                                                        OBSN1040
   44 M=1                                                             OBSN1050
      OBST(J,1) = HV(J)                                               OBSN1060
      OBST(J,2) = 0.                                                  OBSN1070
      OBST(J,3) = 0.                                                  OBSN1080
      GO TO 40                                                        OBSN1090
   46 M=2                                                             OBSN1100
      OBST(J,1) = HA(J)                                               OBSN1110
```

```
      OBST(J,2) = HE(J)                                          OBSN1120
      OBST(J,3) = 0.                                             OBSN1130
      GO TO 40                                                   OBSN1140
   48 M=2                                                        OBSN1150
      OBST(J,1) = HR(J)                                          OBSN1160
      OBST(J,2) = HV(J)                                          OBSN1170
      OBST(J,3) = 0.                                             OBSN1180
      GO TO 40                                                   OBSN1190
   50 M=3                                                        OBSN1200
      OBST(J,1) = HR(J)                                          OBSN1210
      OBST(J,2) = HA(J)                                          OBSN1220
      OBST(J,3) = HE(J)                                          OBSN1230
      GO TO 40                                                   OBSN1240
   52 M=3                                                        OBSN1250
      OBST(J,1) = HV(J)                                          OBSN1260
      OBST(J,2) = HA(J)                                          OBSN1270
      OBST(J,3) = HE(J)                                          OBSN1280
   40 CONTINUE                                                   OBSN1290
      RETURN                                                     OBSN1300
      END                                                        OBSN1310
```

SUBROUTINE MEASUR

COMMON VARIABLES

COMMON BLOCK // ORIGIN 00001 LENGTH 01007

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DATA	00000	R	DATA	00000	R	CON	00000	R
SAT	00017	R	SDA	00043	R	STT	00435	R
WRK	00606	R	OMEGA	00006	R	ITYPE	00705	I
RTRAK	00744	R	X	00761	R	Y	01001	R
Z	01004	R	RVEC	00764	R	VVEC	00767	R
OBST	00501	R						

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
RHO	01010	R	UNIT	01013	R	HR	01016	R
XN	01024	R	VTRAK	01027	R	VREL	01032	R
HV	01035	R	HA	01043	R	HE	01051	R

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
RHOM	01057	R	RHODOT	01060	R	F	01061	R
G	01062	R	S2	01063	R	S	01064	R

ENTRY POINTS

MEASUR SECTION 4

SUBROUTINES CALLED

OBSN STORAGE MAP

AMAG	SECTION	5	CROSS	SECTION	6	DOT	SECTION	7
SQRT	SECTION	8	.FXEM.	SECTION	9	SYSLOC	SECTION	1(

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATIO
1	5A	01101	2	15A	01114	10	32A	01141
3	28A	01136	20	58A	01222	11	40A	01166
12	53A	01216	30	85A	01325	21	69A	01270
22	81A	01322	40	137A	01420	42	90A	01341
44	97A	01350	46	104A	01357	48	112A	01367
50	120A	01377	52	129A	01410			

DECK LENGTH IN OCTAL IS 00446.

2.4.1.4 Subroutine ERROR

Purpose:	To compute the weighting matrix (Q) for the data point (one of 6 types) from uncertainty data provided pertaining to station locations errors and recording instrument errors
Deck Name:	EROR
Calling Sequence:	CALL ERROR (NUMB, M)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Description
I	NUMB	-	1	ARG	Number of the station - this number corresponds to the sequence established in INPUT (comes from TRAK)
I	M	-	1	ARG	Number of pieces of information included in the observation vector (from MEASUR)
I	RE	R_e	1	CON(1)	Earth's equatorial radius (Km)
I	RPOL	R_p	1	CON(2)	Earth's polar radius (Km)
I	OMEGA	ω	1	CON(7)	Earth's spin rate (rad/sec)
I	STERR	$\sigma^2_{\Delta L}, \sigma^2_{\Delta\lambda}, \sigma^2_{\Delta H}$	3	SDA(51)	Variance and latitude, longitude and altitude for the station
I	SNOISE	$\sigma^2_{\Delta\rho}, \sigma^2_{\Delta\dot{\rho}}, \sigma^2_{\Delta A}, \sigma^2_{\Delta E}$	M	SDA(141)	Variances on the observed data for the equipment utilized
I	OBST	$M^T(t)$	6 X 3	STT(37)	Output of MEASUR
O	Q	$Q(t)$	3 X 3	STT(59)	Combined weighting matrix
I	ITYPE	-	1	WRK(64)	Type of data being processed

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Description
I	SLAT	L	1	WRK(105)	Geodetic latitude of the station (rad)
I	SLON	λ	1	WRK(106)	Longitude of the station (rad)
I	H	H	1	WRK(107)	Altitude relative to reference ellipsoid (Km)
I	X (Y,Z)	U (E,N)	3 (3,3)	WRK(108)	Up (East, North) unit vector at station

Subroutines required: MATMPY (matrix multiplication)
TRANSP (matrix transpose)

Functions required: SIN
COS
SQRT

Approximate Deck Length: 627 (octal)

Formulation:

In order to construct the matrix of uncertainty required by the Kalman filter resulting from station position errors and errors in the observations, two assumptions are necessary. The first is that the groups of errors from the two sources under analysis are statistically independent, and the second is that the equations describing the processes are linear. Under these assumptions the resultant uncertainty matrix (Q) is

$$Q = Q_1 + Q_2$$

and attention can be directed to each source independently.

In order to define Q_1, it is first necessary to consider the manner in which the specified set of errors affect the problem (consistent with the assumptions previously outlined). This relationship is

$$\Delta(OBS) = \left[\frac{\partial(OBS)}{\partial L, \lambda, H}\right] \begin{Bmatrix} \Delta L \\ \Delta\lambda \\ \Delta H \end{Bmatrix}$$

$$\equiv [\text{COEFF}] \begin{Bmatrix} \Delta L \\ \Delta \lambda \\ \Delta H \end{Bmatrix}$$

which leads directly to

$$E\left(\Delta(OBS)\,\Delta^T(OBS)\right) = [\text{COEFF}]\; E\left(\begin{Bmatrix} \Delta L \\ \Delta \lambda \\ \Delta H \end{Bmatrix} \{\Delta L\; \Delta\lambda\; \Delta H\}\right)\; [\text{COEFF}]$$

$$Q_1 \equiv [\text{COEFF}]\;\; \sigma\;\; [\text{COEFF}]$$

where σ is specified.

The problem to which this analysis must address itself is thus the construction of the matrix [COEFF] . This development is simplified considerably, since the chain rule can be employed to produce the desired results as follows:

$$\text{COEFF} = \left[\frac{\partial(\overrightarrow{OBS})}{\partial\ \vec{X}}\right]\left[\frac{\partial \vec{X}}{\partial L, \lambda, H}\right] = M\left[\frac{\partial \vec{X}}{\partial L, X, H}\right]$$

$$\equiv [OBS][DERIV]$$

where $[\text{OBS}]^T$ is the output of subroutine MEASUR and where $\vec{X}$ is the state vector. For this reason, subsequent analyses will be directed to the derivation of the matrix DERIV.

The first steps in relating uncertainties in the position and velocity of the tracking equipment and the corresponding uncertainty in the satellite's position and velocity are the definition of the observed quantities

$$\vec{r}_r = \vec{r} - \vec{r}_r$$

$$\vec{V}_r = \vec{V} - \vec{V}_T \quad ,$$

where: the subscripts r and t denote relative and tracking station, respectively, and the description of a coordinate system in which the vectors will be defined (in this case the true equator of date frame).

$$\vec{r}_T = \left(\frac{R_e}{C} + H\right)\left\{\cos L \cos A\, \hat{x} + \cos L \sin A\, \hat{y}\right\}$$

$$+ \left(\frac{R_p^2}{R_e C} + H\right)\left\{\sin L\, \hat{z}\right\}$$

$$\vec{V}_T = \omega\left(\hat{z} \times \vec{r}_T\right)$$

$$\vec{V}_T = \omega\left(\frac{R_e}{C} + H\right)\left[-\cos L \sin A\,\hat{X} + \cos L \cos A\,\hat{Y}\right]$$

$$= -\omega Y_T\,\hat{X} + \omega X_T\,\hat{Y} + 0\,\hat{Z}$$

where R_e = equatorial radius of the oblate spheroid
R_p = polar radius of the oblate spheroid
c^2 = a function of latitude
$= \cos^2 L + (R_p/R_e)^2 \sin^2 L$
L = geodetic latitude of the station
A = right ascension of the station
= longitude relative to Greenwich (λ) plus sidereal time of the Greenwich meridian
ω = spin rate of the earth
$\hat{X}$ = unit vector toward the vernal equinox
$\hat{Z}$ = unit vector along the spin vector of the earth

Now if the vehicle's position and velocity are held constant for the purposes of differentiation, the sensitivity of the state vector to errors in the tracking stations position and velocity can be obtained by computing the partials of the relative $\vec{r}$ and $\vec{v}$ with respect to errors in L, A, and H

$$\frac{\partial \vec{r}}{\partial u} = -\frac{\partial \vec{r}_T}{\partial u} \qquad u = L, A, H$$

$$\frac{\partial \vec{V}}{\partial u} = -\frac{\partial \vec{V}_T}{\partial u} \qquad u = L, A, H$$

However, before these derivatives are evaluated, it is noted that since right ascension can be expressed as

$$A = \omega(t - t_0) + \lambda$$

and since sidereal time is uncertain to a degree much less than that associated with λ , $dA \approx d\lambda$.

Now

$$\frac{\partial X_T}{\partial L} = -\left(\frac{R_e}{C} + H\right)\sin L \cos A + \cos L \cos A\left(-\frac{R_e}{C^2}\right)\frac{\partial C}{\partial L}$$

$$\frac{\partial X_T}{\partial \lambda} = -\left(\frac{R_e}{C} + H\right)\cos L \sin A$$

$$\frac{\partial X_T}{\partial H} = \cos L \cos A$$

$$\frac{\partial Y_T}{\partial L} = -\left(\frac{R_e}{C} + H\right) \sin L \sin A + \cos L \sin A \left(\frac{-R_e}{C^2}\right) \frac{\partial C}{\partial L}$$

$$\frac{\partial Y_T}{\partial \lambda} = \left(\frac{R_e}{C} + H\right) \cos L \cos A$$

$$\frac{\partial Y_T}{\partial H} = \cos L \sin A$$

$$\frac{\partial Z_T}{\partial L} = \left(\frac{R_p^2}{C R_e} + H\right) \cos L + \left(\frac{-R_p^2}{R_e C^2}\right) \frac{\partial C}{\partial L}$$

$$\frac{\partial Z_T}{\partial \lambda} = 0$$

$$\frac{\partial Z_T}{\partial H} = \sin L$$

where

$$\frac{1}{C^2} \frac{\partial C}{\partial L} = \frac{-\sin L \cos L}{C^3} \left[1 - \left(\frac{R_p}{R_e}\right)^2\right]$$

$\equiv$ FUNCT (in program language)

and

$$\frac{\partial \dot{X}_T}{\partial L} = -\omega \frac{\partial Y_T}{\partial L}$$

$$\frac{\partial \dot{X}_T}{\partial \lambda} = -\omega \frac{\partial Y_T}{\partial \lambda}$$

$$\frac{\partial \dot{X}_T}{\partial H} = -\omega \frac{\partial Y_T}{\partial H}$$

$$\frac{\partial \dot{Y}_T}{\partial L} = +\omega \frac{\partial X_T}{\partial L}$$

$$\frac{\partial \dot{Y}_T}{\partial \lambda} = +\omega \frac{\partial X_T}{\partial \lambda}$$

$$\frac{\partial \dot{Y}_T}{\partial H} = +\omega \frac{\partial X_T}{\partial H}$$

$$\frac{\partial \dot{Z}_T}{\partial u} = 0 \qquad u = L, \lambda, H$$

With the evaluation of DERIV, the array COEFF is known and Q_1 can be computed once data for the errors in the variables L, λ, and H are specified. Attention thus turns to the second source of errors to be considered; namely, those resulting from observational uncertainties. Assuming that the observation vector for this analysis is ordered in the same fashion

as for the previous analysis, Q_2 is by definition

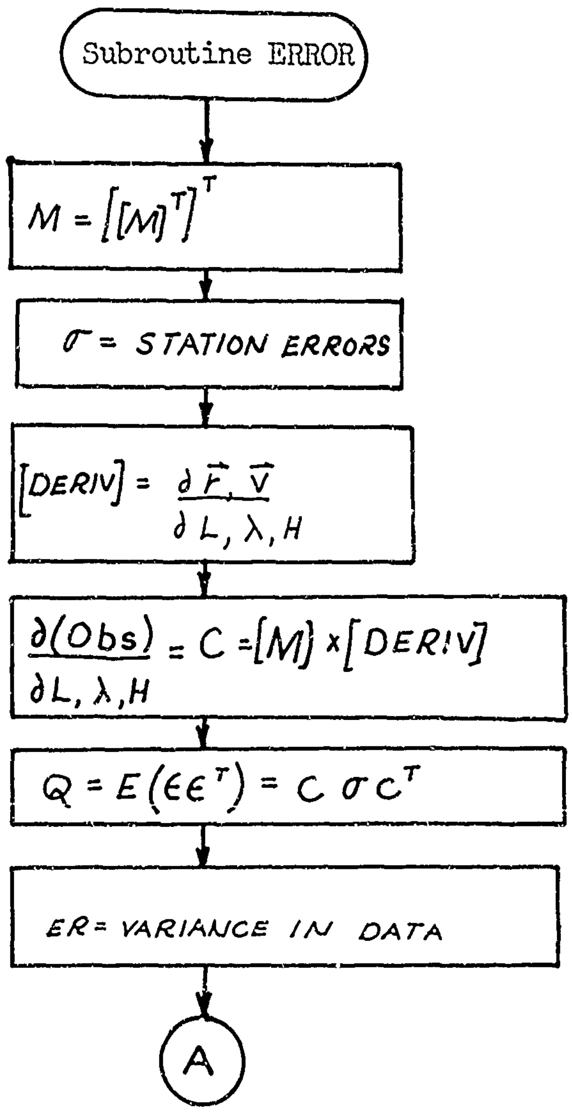
$$Q_2 = E(\epsilon \epsilon^T)$$

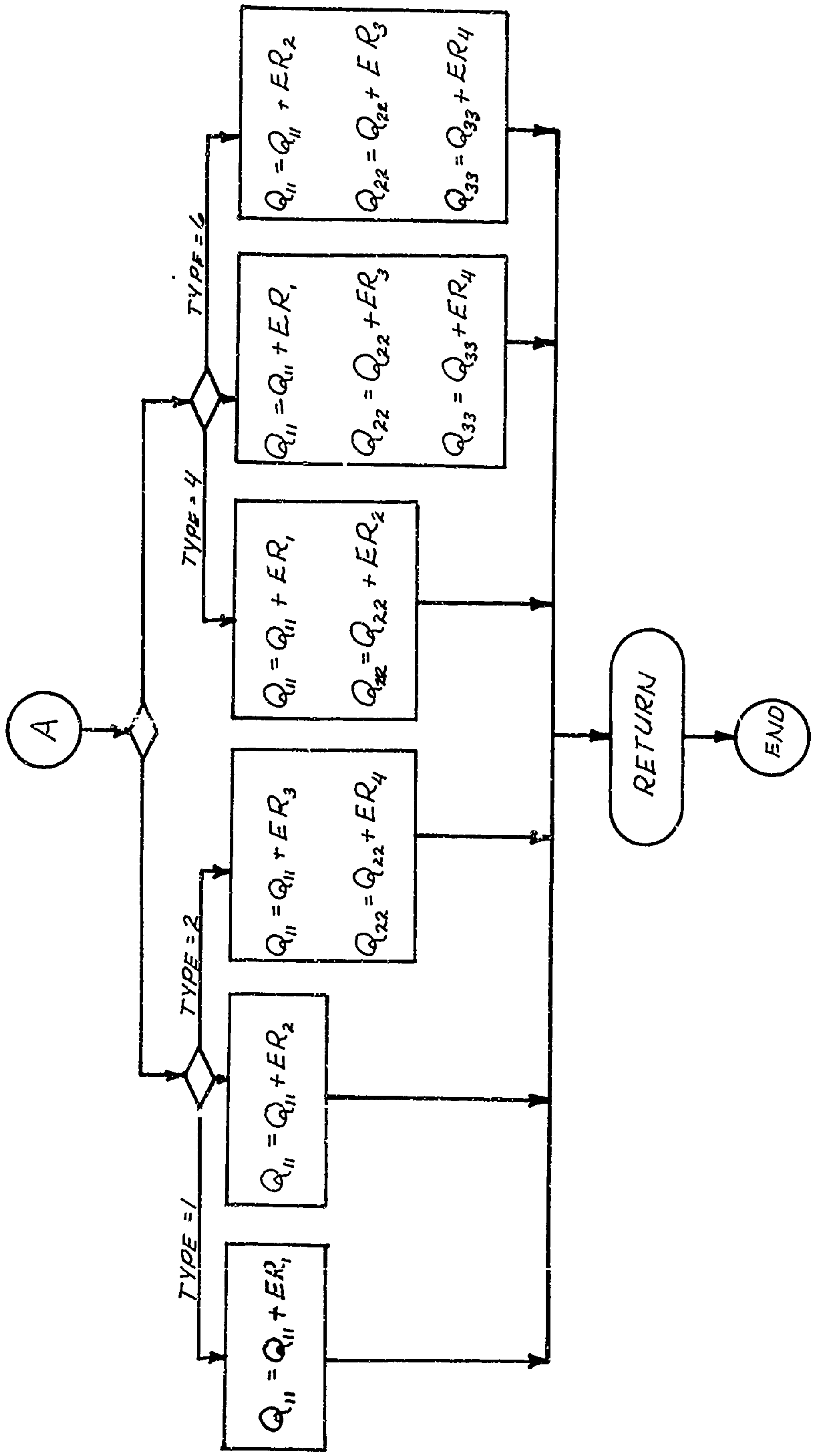
$$\epsilon = Y_{TRUE} - Y_{OBSERVED}$$

and is provided as input data for each station.

For the present, both σ and Q_2 have been assumed to be diagonal (i.e., the errors involved are uncorrelated); and provision has been made in subroutine INPUT to store only the variances. However, should station calibration and survey indicate that a significant degree of correlation exists between the elements internal to these arrays, other terms must be added. This step requires only the modification of INPUT and ERROR.

Computational Logic:




A
TYPE = 1
$Q_{11} = Q_{11} + ER_1$
$Q_{11} = Q_{11} + ER_2$
TYPE = 2
$Q_{11} = Q_{11} + ER_3$
$Q_{22} = Q_{22} + ER_4$
TYPE = 4
$Q_{11} = Q_{11} + ER_1$
$Q_{22} = Q_{22} + ER_2$
$Q_{11} = Q_{11} + ER_1$
$Q_{22} = Q_{22} + ER_3$
$Q_{33} = Q_{33} + ER_4$
TYPE = 6
$Q_{11} = Q_{11} + ER_2$
$Q_{22} = Q_{22} + ER_3$
$Q_{33} = Q_{33} + ER_4$
RETURN
END

```
      SUBROUTINE ERROR(NUMB,M)                                          EROR0020
C                                                                       EROR0030
C     THIS ROUTINE IS DESIGNED TO COMPUTE THE COVARIANCE MATRIX FOR     EROR0040
C              THE ERRORS IN THE OBSERVATION MATRIX DUE TO STATION      EROR0050
C              LOCATION INACCURACIES . IT THEN CONSTRUCTS THE COMPLETE  EROR0060
C              COVARIANCE MATRIX * Q * OF THE KALMAN FILTER BY ADDING   EROR0070
C              THE STATION ERROR MATRIX TO THE COVARIANCE MATRIX FOR    EROR0080
C              THE NOISE IN THE MEASUREMENTS .                          EROR0090
C     SLAT  = STATION LATITUDE IN RADIANS                               EROR0100
C     SLON  = STATION LONGITUDE IN RADIANS                              EROR0110
C     H     = STATION ALTITUDE RELATIVE TO ELLIPSOID IN UNITS OF RE,RPOLEROR0120
C     X,Y,Z = UNIT VECTORS (UP,EAST,NORTH) WITH ORIGIN AT THE STATION   EROR0130
C     ITYPE = DATA TYPE ( SEE ROUTINE MEASUR FOR OPTIONS )              EROR0140
C     NUMB  = TRACKING STATION NUMBER (DEPENDS ON ARRANGEMENT OF        EROR0150
C              LATITUDE,LONGITUDE,AND ALTITUDE DATA IN COMMON STARTING  EROR0160
C              AT DATA(51) AND GOING UPTO DATA(92) )                    EROR0170
C     M     = DIMENSIONALITY OF THE OBSERVATION VECTOR                  EROR0180
C                                                                       EROR0190
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * EROR0200
C                                                                       EROR0210
      DIMENSION  CON(1), SAT(1), SDA(1), WRK(1), STT(1)                 EROR0220
C                                                                       EROR0230
      COMMON DATA                                                       EROR0240
C                                                                       EROR0250
      EQUIVALENCE  (DATA(  1),CON), (DATA( 16),SAT), (DATA( 36),SDA)    EROR0260
     2            ,(DATA(286),STT), (DATA(391),WRK)                     EROR0270
C                                                                       EROR0280
      DIMENSION  X(3)       ,Y(3)       ,Z(3)       ,STERR(90) ,SNOISE(100)EROR0290
     2,OBST(6,3),Q(3,3)     ,DERIV(6,3),SIGMA(3,3),ER(4)      ,OBS(3,6)  EROR0300
     3,COEFF(3,3),COEFT(3,3),DUMMY(3,3)                                 EROR0310
C                                                                       EROR0320
      EQUIVALENCE          (CON(  1),RE     ), (CON(  2),RPOL  )        EROR0330
     2                    ,(CON(  7),OMEGA )                            EROR0340
      EQUIVALENCE          (WRK( 64),ITYPE ), (WRK(105),SLAT  )         EROR0350
     2  ,(WRK(106),SLON  ), (WRK(107),H     ), (WRK(108),X     )        EROR0360
     3  ,(WRK(124),Y     ), (WRK(127),Z     )                           EROR0370
```

```
      EQUIVALENCE          (SDA( 51),STERR ), (SDA(141),SNOISE)     EROR0380
      EQUIVALENCE          (STT( 37),OBST  ), (STT( 55),Q     )     EROR0390
C                                                                   EROR0400
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   EROR0410
C                                                                   EROR0420
      CALL TRANSP(OBST,6,3,OBS)                                     EROR0430   2
      DO 1 I=1,3                                                    EROR0440
      DO 1 J=1,3                                                    EROR0450
      IF (I-J) 2,3,2                                                EROR0460
    2 SIGMA(I,J) = 0.                                               EROR0470
      GO TO 1                                                       EROR0480
    3 SIGMA(I,J) = STERR(3*NUMB-2)                                  EROR0490
    1 CONTINUE                                                      EROR0500
C                                                                   EROR0510
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * EROR0520
C                                                                   EROR0530
C     THE FIRST STEP WILL BE TO EVALUATE THE MATRIX OF PARTIAL      EROR0540
C              DERIVATIVES OF THE STATE VECTOR WITH RESPECT TO THE  EROR0550
C              POSITION COORDINATES OF THE TRACKING STATION ( THE   EROR0560
C              ASSUMED ORDER FOR THE POSITION VECTOR IS LATITUDE ,  EROR0570
C              LONGITUDE , AND ALTITUDE  ).  * * * * * * * * * * * * EROR0580
C                                                                   EROR0590
      SLA= SIN(SLAT)                                                EROR0600   20
      CLA= COS(SLAT)                                                EROR0610   21
      SLN= SIN(SLON)                                                EROR0620   22
      CLN= COS(SLON)                                                EROR0630   23
      C  = SQRT( CLA*CLA + RPOL*RPOL*SLA*SLA/(RE*RE))               EROR0640   24
      AOC= RE/C                                                     EROR0650
      CIN= 1./C                                                     EROR0660
      RATIO=RPOL*RPOL/RE                                            EROR0670
      FUNCT =-CIN *(RE-RATIO)* SLA*CLA*CIN*CIN /RE                  EROR0680
C                                                                   EROR0690
      DERIV(1,1) = RE*FUNCT*X(1) - (AOC+ H)*Z(1)                    EROR0700
      DERIV(1,2) =-(AOC+ H)*CLA*Y(1)                                EROR0710
      DERIV(1,3) =-X(1)                                             EROR0720
      DERIV(2,1) = RE*FUNCT*X(2) - (AOC+ H)*Z(2)                    EROR0730
      DERIV(2,2) =-(AOC+ H)*CLA*Y(2)                                EROR0740
```

```
      DERIV(2,3) =-X(2)                                                 EROR0750
      DERIV(3,1) = RATIO*FUNCT*X(3) - (RATIO/C +H)*Z(3)                 EROR0760
      DERIV(3,2) = 0.                                                   EROR0770
      DERIV(3,3) =-X(3)                                                 EROR0780
      DERIV(4,1) =-OMEGA * DERIV(2,1)                                   EROR0790
      DERIV(4,2) =-OMEGA * DERIV(2,2)                                   EROR0800
      DERIV(4,3) =-OMEGA * DERIV(2,3)                                   EROR0810
      DERIV(5,1) =+OMEGA * DERIV(1,1)                                   EROR0820
      DERIV(5,2) =+OMEGA * DERIV(1,2)                                   EROR0830
      DERIV(5,3) =+OMEGA * DERIV(1,3)                                   EROR0840
      DERIV(6,1) = 0.                                                   EROR0850
      DERIV(6,2) = 0.                                                   EROR0860
      DERIV(6,3) = 0.                                                   EROR0870
C                                                                       EROR0880
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * EROR0890
C                                                                       EROR0900
C     AT THIS POINT THE ERROR MATRIX RELATING THE VARIATION IN THE      EROR0910
C              OBSERVABLES TO THOSE IN THE STATION LOCATION CAN BE      EROR0920
C              COMPUTED AND THE EXPECTED VALUE MATRIX Q DETERMINED .    EROR0930
C                                                                       EROR0940
      CALL MATMPY( OBS ,3,6,DERIV,6,3,COEFF )                           EROR0950    25
      CALL TRANSP( COEFF,3,3,COEFT )                                    EROR0960    27
      CALL MATMPY( COEFF,3,3,SIGMA,3,3,DUMMY)                           EROR0970    29
      CALL MATMPY( DUMMY,3,3,COEFT,3,3, Q )                             EROR0980
C                                                                       EROR0990
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * EROR1000
C                                                                       EROR1010
C     AT THIS POINT THE STATION NOISE CHARACTERISTICS WILL BE ADDED.    EROR1020    31
      DO 10 I=1,4                                                       EROR1030
   1  ER(I)=SNOISE( 4*NUMB -3 )                                         EROR1040
      GO TO(20,30,40,50,60,70 ),ITYPE                                   EROR1050
   2  Q(1,1) =ER(1) +Q(1,1)                                             EROR1060
      GO TO 80                                                          EROR1070
  30  Q(1,1) =ER(2) +Q(1,1)                                             EROR1080
      GO TO 80                                                          EROR1090
  40  Q(1,1) =ER(3) +Q(1,1)                                             EROR1100
      Q(2,2) =ER(4) +Q(2,2)                                             EROR1110
```

```
      GO TO 80                                            EROR1120
   50 Q(1,1) =ER(1) +Q(1,1)                               EROR1130
      Q(2,2) =ER(2) +Q(2,2)                               EROR1140
      GO TO 80                                            EROR1150
   60 Q(1,1) =ER(1) +Q(1,1)                               EROR1160
      Q(2,2) =ER(3) +Q(2,2)                               EROR1170
      Q(3,3) =ER(4) +Q(3,3)                               EROR1180
      GO TO 80                                            EROR1190
   70 Q(1,1) =ER(2) +Q(1,1)                               EROR1200
      Q(2,2) =ER(3) +Q(2,2)                               EROR1210
      Q(3,3) =ER(4) +Q(3,3)                               EROR1220
   80 RETURN                                              EROR1230
      END                                                 EROR1240
```

STORAGE MAP

SUBROUTINE ERROR

COMMON VARIABLES

COMMON BLOCK // ORIGIN 00001 LENGTH 01007

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DATA	00000	R	DATA	00000	R	CON	00000	R
SAT	00017	R	SDA	00043	R	STT	00435	R
WRK	00406	R	RE	00000	R	RPOL	00001	R
OMEGA	00006	R	ITYPE	00705	I	SLAT	00756	R
SLON	00757	R	H	00760	R	X	00761	R
Y	01001	R	Z	01004	R	STERR	00125	R
SNOISE	00257	R	OBST	00501	R	Q	00523	R

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DERIV	01010	R	SIGMA	01032	R	ER	01043	R
OBS	01047	R	COEFF	01071	R	COEFT	01102	R
DUMMY	01113	R						

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
I	01124	I	J	01125	I	SLA	01126	R
CLA	01127	R	SLN	01130	R	CLN	01131	R
C	01132	R	AOC	01133	R	CIN	01134	R
RATIO	01135	R	FUNCT	01136	R			

ENTRY POINTS

ERROR SECTION 4

SUBROUTINES CALLED

TRANSP	SECTION	5	SIN	SECTION	6	COS	SECTION	7
SORT	SECTION	8	MATMPY	SECTION	9	.FXEM.	SECTION	10
SYSLOC	SECTION	1						

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1	16A	01210	2	10A	01204	3	13A	01206
10	36A	01511	20	42A	01530	30	44A	01534
40	46A	01540	50	48A	01547	60	50A	01556
70	52A	01570	80	53A	01601			

DECK LENGTH IN OCTAL IS 00626.

2.4.1.5 Subroutine UPSTAT

Purpose: To determine if the uncertainties in the estimate of the state of the system are sufficiently small to justify updating the position (velocity) by adding the state vector to the position and velocity vectors on the nominal trajectory and to write the results of the data reduction problem.

Deck Name: UPST

Calling Sequence: CALL UPSTAT

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Description
I/O	STATE	$\vec{X}(t)$	6	STT(64)	The state vector for the system (Km, Km/sec.)
I	P	$P(t)$	6 X 6	STT(70)	Covariance matrix for estimation errors in $\bar{X}(t)$
I	ROTINV	-	3 X 3	WRK(10)	Matrix relating coordinate frame of date to frame of 1950.0
O	RCONIC	$\vec{r}_c, \vec{v}_c$	3,3	WRK(28) WRK(31)	Position and velocity vectors for new conic reference trajectory (Km, Km/sec.)
O	TCONW TCONF	t_c	1,1	WRK(34) WRK(35)	Time in days at which rectification of reference conic occurred (days relative to 1950.0)
O	RTRANS VTRANS	$\vec{r}_o, \vec{v}_o$	3,3	WRK(36) WRK(39)	Position and velocity vectors in frame of date on true trajectory at data point just reduced (Km, Km/sec.)
O	TTRANW, TTRANF	t_t	1,1	WRK(42) WRK(43)	Time from which errors will propagate for next pass through filter (days relative to 1950.0)

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Description
I	TW, TF	t	1,1	WRK(50) WRK(51)	Whole and fractional part of the present day relative to the epoch of 1950.0 (J.D. 2433282.423)
I/O	RDATE, VDATE	$\vec{r}$ $\vec{v}$	3 3	WRK(111) WRK(114)	Position and velocity vectors in the frame of date (Km, Km/sec.)
I	RE GM	R_e GM	1 1	CON(1) CON(6)	Equatorial radius and gravitational constant for the Earth (Km, Km^3/sec^2)
I	NOUT	-	1	CON(14)	Output tape number

Subroutines required: MATMPY (matrix multiplication)
ELEMEN (computes conic elements)

Functions required: DOT (dot product)

Approximate Deck Length: 704 (octal)

Discussion:

UPSTAT (Update State Vector) is designed to determine whether the uncertainty in the estimate of the state vector is sufficiently small to allow the trajectory to be discreetly changed by adding the state vector to the position and velocity vectors on the integrated trajectory. This test is made to assure that large uncertainties in the estimate early in time will not produce the degenerate solution which is possible if noisy estimates of the state are allowed to be introduced directly into the system along with the desired results of the filtering and corrections for the non-linear nature of the problem.

In order to produce this test, however, it is necessary to define a comparison function and specify a measure of error which is considered acceptable as a threshold for updating. This process, quite arbitrarily, has been performed as follows:

$$F = \sum_{i=1}^{3} \left[\left(\frac{\Delta r_i}{r}\right)^2 + \lambda \left(\frac{\Delta v_i}{v}\right)^2 \right]$$

and the weighting factor has been established so that the sensitivity of the semi-major axis to errors in radius and velocity for a circular orbit are equal, i.e.,

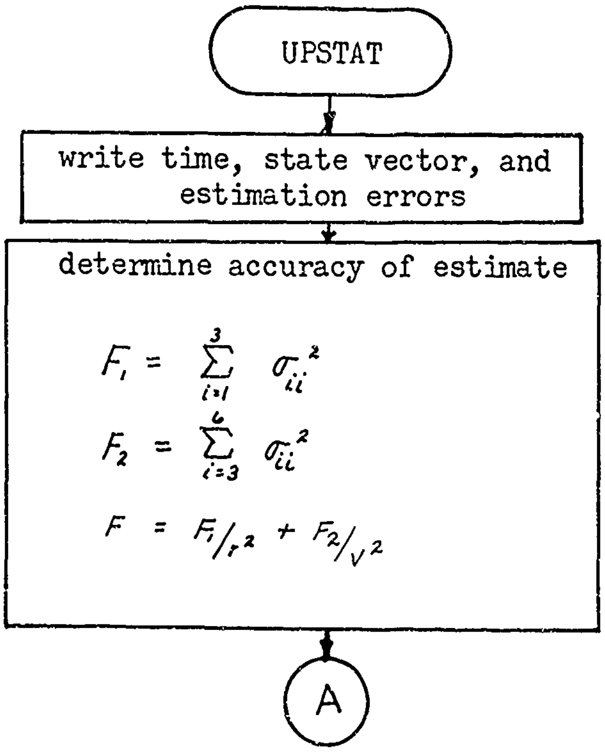
$$\frac{\Delta a_1}{a} = 2\,\frac{a}{r}\,\frac{\Delta r}{r} = 2\,\frac{\Delta r}{r}$$

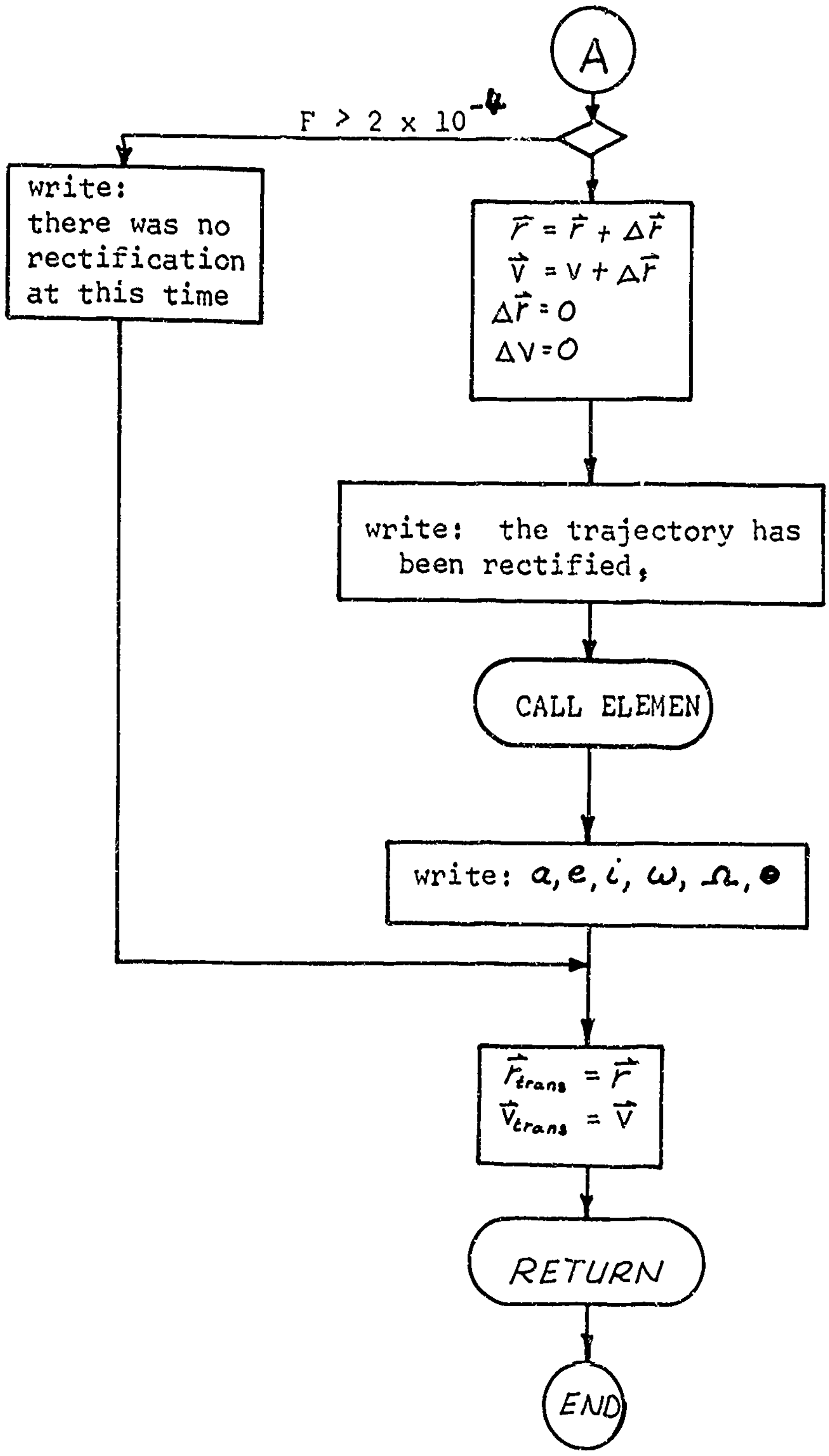
$$\frac{\Delta a_2}{a} = 2\left(2\frac{a}{r} - 1\right)\frac{\Delta V}{V} = 2\ \frac{\Delta V}{V}$$

Thus, for this criteria, the two ratios in question should be weighted equally. That is, $\lambda = 1$. The threshold for this function has been set at 2×10^{-4}.

Upon entry into UPSTAT, the covariance matrix for the errors in the state vector is examined; and the elements along the principle diagonal (the trace) are extracted and utilized as Δr_i (or ΔV_i) in the construction of the comparison function. The test is then made to determine whether updating will be accomplished; if not, the position and velocity vectors at this time are stored for future computation of the transition matrix, and control is returned to the calling routine. However, if updating is practical, it is performed, the conic reference trajectory is rectified, the state vector is zeroed, and the "classic" elements of the osculating conic trajectory are computed for reference (these variables are not utilized in the program due to problems of indeterminancies as e and ω approach zero). Operation from this point is identical to the case where the test was failed.

Computational Logic:




A
F > 2 x 10^-4
write:
there was no
rectification
at this time
r = r + Δr
V = V + Δr
Δr = 0
ΔV = 0
write: the trajectory has
been rectified,
CALL ELEMEN
write: a, e, i, ω, Ω, θ
r_trans = r
V_trans = V
RETURN
END

```
      SUBROUTINE UPSTAT                                                 UPST0020
C                                                                       UPST0030
C     UPSTAT IS DESIGNED TO ALTER THE REFERENCE TRAJECTORY WHEN THE     UPST0040
C             STATE VECTOR IS KNOWN TO REASONABLE PRECISSION . IT IS    UPST0050
C             ALSO DESIGNED TO WRITE OUT THE STATE INFORMATION AND TO   UPST0060
C             STORE THE POSITION AND VELOCITY ARRAYS AT THE TIME OF     UPST0070
C             DATA REDUCTION SO THAT SUBSEQUENT STATE TRANSITION        UPST0080
C             MATRICES CAN BE COMPUTED.                                 UPST0090
C                                                                       UPST0100
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * UPST0110
C                                                                       UPST0120
      DIMENSION  CON(1), SAT(1), SDA(1), WRK(1), STT(1)                 UPST0130
C                                                                       UPST0140
      COMMON DATA                                                       UPST0150
C                                                                       UPST0160
      EQUIVALENCE  (DATA(  1),CON), (DATA( 16),SAT), (DATA( 36),SDA)    UPST0170
     2             ,(DATA(286),STT), (DATA(391),WRK)                    UPST0180
C                                                                       UPST0190
      DIMENSION  ROTINV(3,3),RCONIC(3)  ,VCONIC(3)  ,RTRANS(3)          UPST0200
     2,VTRANS(3),RDATE(3)   ,VDATE(3)   ,STATE(6)   ,P(6,6)             UPST0210
     3 ,B(4)     ,RCOND(3)   ,VCOND(3)   ,RR(3)      ,VV(3)             UPST0220
C                                                                       UPST0230
      EQUIVALENCE            (CON(  1),RF     ), (CON(  6),GM     )     UPST0240
     2                      ,(CON( 14),NOUT   )                         UPST0250
      EQUIVALENCE            (WRK( 50),TW     ), (WRK( 51),TF     )     UPST0260
      EQUIVALENCE            (WRK( 10),ROTINV), (WRK( 28),RCONIC)       UPST0270
     2  ,(WRK( 31),VCONIC), (WRK( 34),TCONW ), (WRK( 35),TCONF )        UPST0280
     3  ,(WRK( 36),RTRANS), (WRK( 39),VTRANS), (WRK( 42),TTRANW)        UPST0290
     4  ,(WRK( 43),TTRANF), (WRK(111),RDATE ), (WRK(114),VDATE )        UPST0300
     5  ,(WRK( 69),TWPRNT), (WRK( 70),TFPRNT)                           UPST0305
      EQUIVALENCE            (STT( 64),STATE ), (STT( 70),P      )      UPST0310
C                                                                       UPST0320
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *     UPST0330
C                                                                       UPST0340
      WRITE(6,1) TW,TF,TWPRNT,TFPRNT                                    UPST0350
    1 FORMAT(1H1,2X,27HEPOCH (REL. TO 1950.0    ) =,2F17.8,1X,12H(DAYS,DAUPST0352
```

```
     UPST        -  EFN   SOURCE STATEMENT  -  IFN(S)  -

  2YS )   /1H ,2X,27HEPOCH (REL. JD 2433282.5) =,2E17.8,1X,12H(DAYS  UUPST0354
  3.T.)   )                                                          UPST0355
   DO 100 II=1,3                                                     UPST0360
   IPLUS3 = II + 3                                                   UPST0362
   RR(II) = RDATE(II) + STATE(II)                                    UPST0364
   VV(II) = VDATE(II) + STATE(IPLUS3)                                UPST0366
100 CONTINUE                                                         UPST0368
   WRITE(6,2) (RR(I),I=1,3), (VV(I),I=1,3), (STATE(I),I=1,6)         UPST0370    14
 2 FORMAT(1H0,2X,27HPOSITION VECTOR ( DATE )  =,3E17.8               UPST0372
  2       /1H ,2X,27HVELOCITY VECTOR ( DATE )  =,3E17.8              UPST0374
  3       /1H ,2X,27HDELTA RADIUS              =,3E17.8              UPST0376
  4       /1H ,2X,27HDELTA VELOCITY            =,3E17.8   )          UPST0378
   WRITE(   6,3)                                                     UPST0400    33
 3 FORMAT(1H-,2X,54HTHE COVARIANCE MATRIX FOR THE ERRORS IN STATE  **UPST0402
  2 P **   )                                                         UPST0404
   WRITE(   6,4) ((P(I,J),J=1,6),I=1,6 )                             UPST0420    34
 4 FORMAT(1H0 /(6E17.8))                                             UPST0422
C                                                                    UPST0440
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * UPST0450
C                                                                    UPST0460
   F1 = P(1,1) + P(2,2) + P(3,3)                                     UPST0470
   F2 = P(4,4) + P(5,5) + P(6,6)                                     UPST0480
   FUNCTN =  F1/ DOT(RDATE,RDATE) + F2/ DOT(VDATE,VDATE)             UPST0490    44    45
   IF( ABS(FUNCTN) -2.E-6 ) 10,10,20                                 UPST0500
10 DO 11 I=1,3                                                       UPST0510
   RDATE(I) = RDATE(I) + STATE(I)                                    UPST0520
   VDATE(I) = VDATE(I) + STATE(I+3)                                  UPST0530
   RCOND(I)= RDATE(I)                                                UPST0540
   VCOND(I)= VDATE(I)                                                UPST0550
   STATE(I) = 0.                                                     UPST0560
11 STATE(I+3) = 0.                                                   UPST0570
   TCONW = TW                                                        UPST0580
   TCONF = TF                                                        UPST0590
   CALL MATMPY(ROTINV,3,3,RCOND,3,1,RCONIC)                          UPST0600    66
   CALL MATMPY(ROTINV,3,3,VCOND,3,1,VCONIC)                          UPST0610    68
   ICONIC = 1                                                        UPST0620
   CALL ELEMEN(RDATE,VDATE,GM,RE,A,E,PP,B(1),B(2),B(3),B(4),XX,YY)   UPST0630    71
```

```
      DO 40 I=1,4                                                       UPST0640
   40 B(I) = B(I)*57.295780                                             UPST0650
      WRITE(6,12) (RDATE(I),I=1,3), (VDATE(I),I=1,3), A, E, (B(I),I=1,4)UPST0660
     2            ,XX, YY                                               UPST0662     80
   12 FORMAT(1H-,2X,33HTHE TRAJECTORY HAS BEEN RECTIFIED                UPST0664
     2       /1H0,2X,27HPOSITION VECTOR (R DATE)  =,3E17.8              UPST0666
     3       /1H ,2X,27HVELOCITY VECTOR (V DATE)  =,3E17.8              UPST0668
     4       /1H-,2X,23HELLIPTIC ORBIT ELEMENTS                         UPST0670
     5       /1H0,2X,27HSEMIMAJOR AXIS       (KM)  =, E17.8             UPST0672
     6       /1H ,2X,27HECCENTRICITY               =, E17.8             UPST0674
     7       /1H ,2X,27HTRUE ANOMOLY DATE    (DEG) =, E17.8             UPST0676
     8       /1H ,2X,27HR ASC OF ASC NODE    (DEG) =, E17.8             UPST0678
     9       /1H ,2X,27HARG OF PERIAPSE      (DEG) =, E17.8             UPST0680
     1       /1H ,2X,27HORBIT INCLINATION    (DEG) =, E17.8             UPST0682
     2       /1H ,2X,27HDEL NODE PER SEC     (RAD) =, E17.8             UPST0684
     3       /1H ,2X,27HDEL APSE PER SEC     (RAD) =, E17.8  /1H1)      UPST0686
      GO TO 30                                                          UPST0760
   20 WRITE(   6,21)                                                    UPST0770     100
   21 FORMAT(1H-,2X,40HTHERE WAS NO RECTIFICATION AT THIS TIME  /1H1)   UPST0780
      ICONIC = 2                                                        UPST0790
   30 DO 31 I=1,3                                                       UPST0800
      RTRANS(I) = RDATE(I)                                              UPST0810
   31 VTRANS(I) = VDATE(I)                                              UPST0820
      TTRANW = TW                                                       UPST0830
      TTRANF = TF                                                       UPST0840
      RETURN                                                            UPST0850
      END                                                               UPST0860
```

STØRAGE MAP

SUBRØUTINE UPSTAT

COMMØN VARIABLES

COMMØN BLØCK // ØRIGIN 00001 LENGTH 00772

SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE
DATA	00000	R	DATA	00000	R	CØN	00000	R
SAT	00017	R	SDA	00043	R	STT	00435	R
WRK	00606	R	RE	00000	R	GM	00005	R
NØUT	00015	I	TW	00667	R	TF	00670	R
RØTINV	00617	R	RCØNIC	00641	R	VCØNIC	00644	R
TCØNW	00647	R	TCØNF	00650	R	RTRANS	00651	R
VTRANS	00654	R	TTRANW	00657	R	TTRANF	00660	R
RDATE	00764	R	VDATE	00767	R	TWPRNT	00712	R
TFPRNT	00713	R	STATE	00534	R	P	00542	R

DIMENSIØNED PRØGRAM VARIABLES

SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE
B	00773	R	RCØND	00777	R	VCØND	01002	R
RR	01005	R	VV	01010	R			

UNDIMENSIØNED PRØGRAM VARIABLES

SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE
II	01013	I	IPLUS3	01014	I	I	01015	I
F1	01016	R	F2	01017	R	FUNCTN	01020	R
ICØNIC	01021	I	A	01022	R	F	01023	R
PP	01024	R	XX	01025	R	YY	01026	R

ENTRY PØINTS

UPSTAT SECTIØN 4

SUBRØUTINES CALLED

.FWRD.	SECTIØN	5	DØT	SECTIØN	6	MATMPY	SECTIØN	7
ELEMEN	SECTIØN	8	.UN06.	SECTIØN	9	.FFIL.	SECTIØN	10
.FCNV.	SECTIØN	11	SYSLØC	SECTIØN	12			

EFN IFN CØRRESPØNDENCE

EFN	IFN	LØCATIØN	EFN	IFN	LØCATIØN	EFN	IFN	LØCATIØN
1	FØRMAT	01043	100	12A	01433	2	FØRMAT	01072
3	FØRMAT	01143	4	FØRMAT	01157	10	47A	01556
20	100A	01713	11	62A	01572	40	76A	01650
12	FØRMAT	01162	30	101A	01723	21	FØRMAT	01364
31	107A	01726						

DECK LENGTH IN ØCTAL IS 00770.

2.4.1.5.1 Subroutine ELEMEN

Purpose: to compute the classical conic elements (for the purpose of user convenience) at those times when estimation of the state vector has been performed (providing that tolerances established in UPSTAT for the estimation errors are satisfied).

Deck Name: ELEM

Calling Sequence: CALL ELEMEN (R, V, GM, REQ, A, E, P, THETA, OMEGA, OMEGAW, OINC, OMDOT, WDOT)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	R	$\vec{r}$	3	ARG	radius vector (cartesian) Km
I	V	$\vec{v}$	3	ARG	velocity vector (cartesian) km/sec
I	GM	u	1	ARG	gravitational constant for central body (km^3/sec^2)
I	REQ	R_e	1	ARG	equatorial radius (km)
0	A	a	1	ARG	semimajor axis (km)
0	E	e	1	ARG	eccentricity
0	P	p	1	ARG	semilatus rectum (km)
0	THETA	θ	1	ARG	true anomaly of epoch (rad)
0	OMEGA	Ω	1	ARG	right ascension of ascending node (rad)
0	OMEGAW	ω	1	ARG	argument of periapse (rad)
0	OINC	i	1	ARG	orbital inclination (rad)
0	OMDOT	$\Delta\Omega$	1	ARG	secular change in Ω due to earths oblateness (rad/rev)
0	WDOT	$\Delta\omega$	1	ARG	secular change in ω due to earths oblateness (rad/rev)

Subroutines required:	CROSS	(cross product)
Functions required:	AMAG DOT ATAN SQRT SIN	(vector magnitude) (dot product) (arc tangent) (square root)
Approximate Deck Length:	640	(octal)

Formulation:

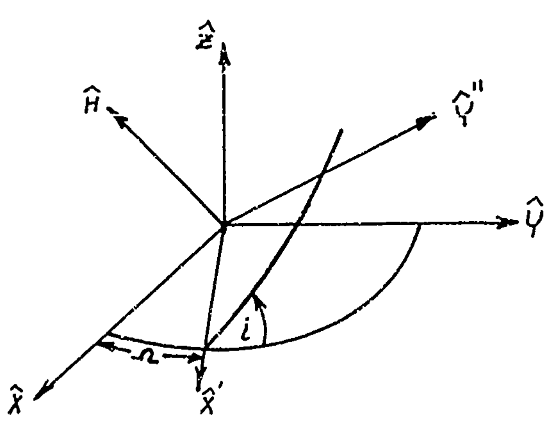
The paragraphs which follow present a summary of the equations which are mechanized in the routine and other information required to resolve ambiguities in quadrant, etc. This approach to the formulation assumes a degree of familiarity with the equations of conic motion in terms of the "classic" elements; thus, should questions arise, a discussion similar to that of Ref. 1 should be reviewed.

The first step to be performed is the definition of the plane of motion. This set of computations will be accomplished through consideration of the angular momentum vector (per unit mass) and will assume that the position and velocity vectors exist in a cartesian format in some selected frame of reference (the main program operates in the true equator of date frame).

$$\vec{h} = \vec{r} \times \vec{V}$$

$$\hat{H} = \vec{h}/|\vec{h}|$$

The orientation of $\hat{H}$ can now be obtained in terms of the orbital inclination and the right ascension of the ascending node by two simple rotational transformations



$$\begin{Bmatrix} \hat{x}' \\ \hat{y}' \\ \hat{z}' \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos i & \sin i \\ 0 & -\sin i & \cos i \end{bmatrix} \cdot \begin{bmatrix} \cos\Omega & \sin\Omega & 0 \\ -\sin\Omega & \cos\Omega & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{Bmatrix}$$

and

$$\hat{H} = \sin i \sin \Omega \hat{X} - \sin i \cos \Omega \hat{Y} + \cos i \hat{Z}$$

Now equating the two expressions for $\hat{H}$ yields the desired information:

$$\cos i = H_3 \qquad 0 < i < 180$$

$$\sin \Omega = H_1 / \sin i$$

$$\cos \Omega = -H_2 / \sin i$$

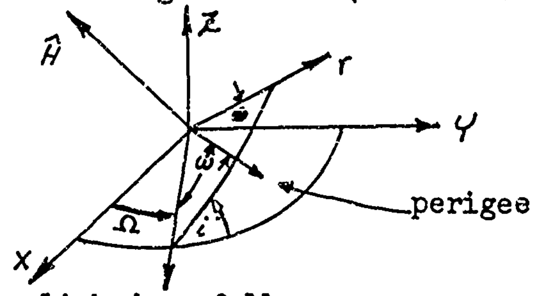
The next step is the computation of the energy of the conic (or its equivalent the semimajor axis) and the eccentricity. This computation involves both the energy and angular momentum equations

$$-\frac{\mu}{2a} = \frac{V^2}{2} - \frac{\mu}{r} \qquad a < 0 \quad \text{Hyperbolic motion}$$

$$p = |\vec{h}|^2/\mu \qquad a > 0 \quad \text{Elliptic motion}$$

$$e = \sqrt{1 - p/a}$$

The final step is the computation of the argument of periapse (ω) and the position in the orbit at the given time (θ = true anomaly of epoch) (see sketch)



This evaluation is accomplished as follows:

$$\varphi = \cos^{-1}\left[\frac{\hat{x}' \cdot \vec{r}}{|\vec{r}|}\right] \qquad \varphi = \text{angle from ascending node} = \theta + \omega$$

$$0 < \varphi < \pi \qquad r_3 > 0$$

$$\pi < \varphi < 2\pi \qquad r_3 < 0$$

$$\theta = \cos^{-1}\left[\frac{p-r}{re}\right] \qquad 0 < \theta < \pi \qquad \vec{r}\cdot\vec{v} > 0$$

$$\pi < \theta < 2\pi \qquad \vec{r}\cdot\vec{v} < 0$$

$$\omega = \varphi - \theta$$

The routine concludes with the computation of the secular perturbations in the right ascension of the ascending node (Ω) and in the argument of periapse (ω) resulting from the first coefficient of the potential function (J_2)

$$\Delta\Omega = -3\pi J_2 \left(\frac{R_e}{p}\right)^2 \cos i \qquad \text{rad/rev}$$

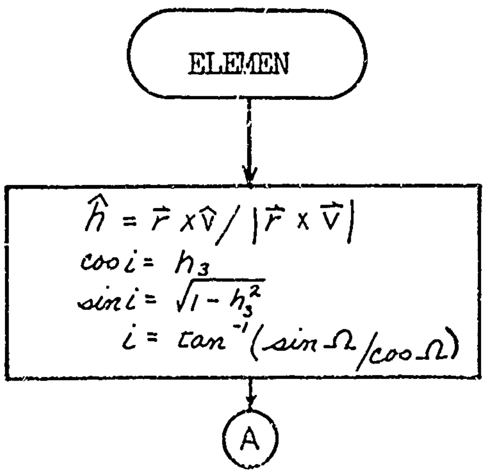
$$\Delta\omega = 3\pi J_2 \left(\frac{R_e}{p}\right)^2 \left(2 - \tfrac{5}{2}\sin^2 i\right) \qquad \text{rad/rev}$$

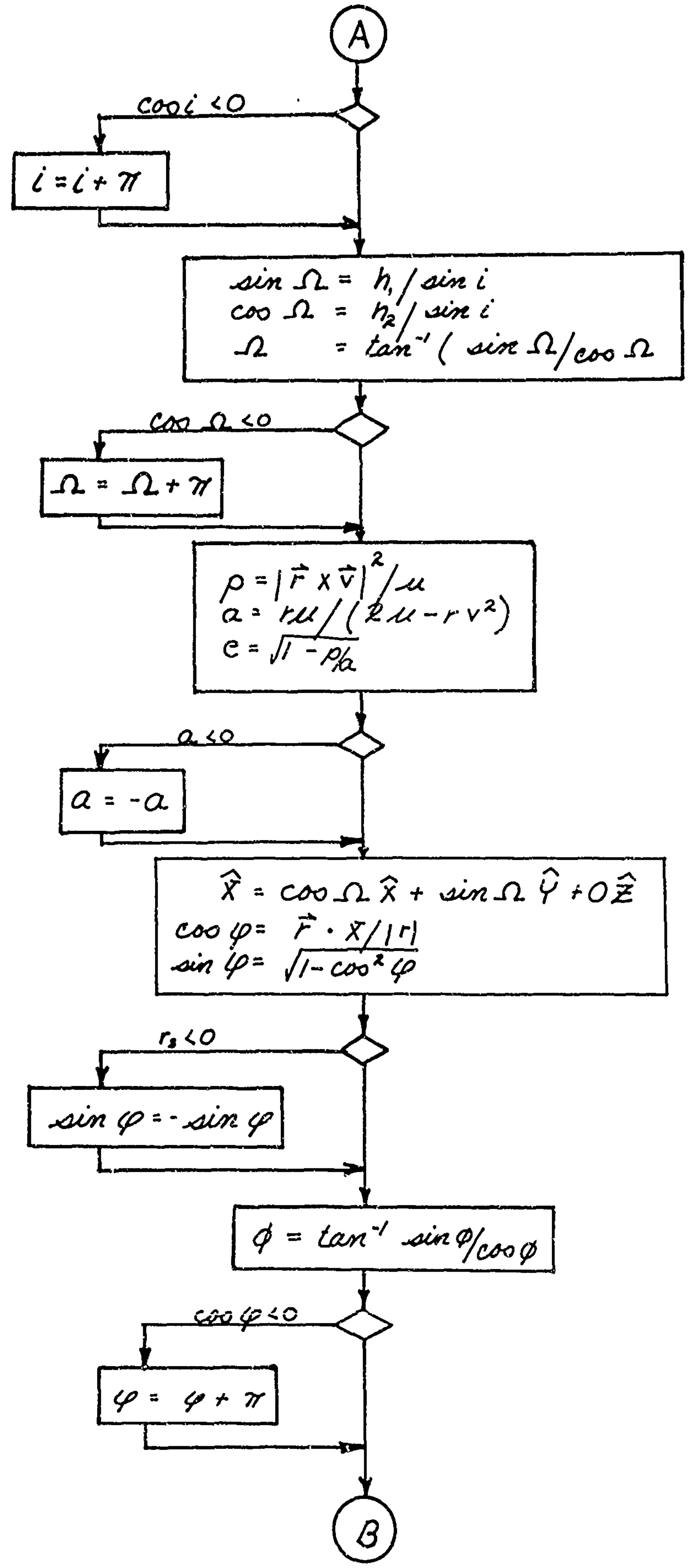
where: R_e = the equatorial radius of the oblate spheroid

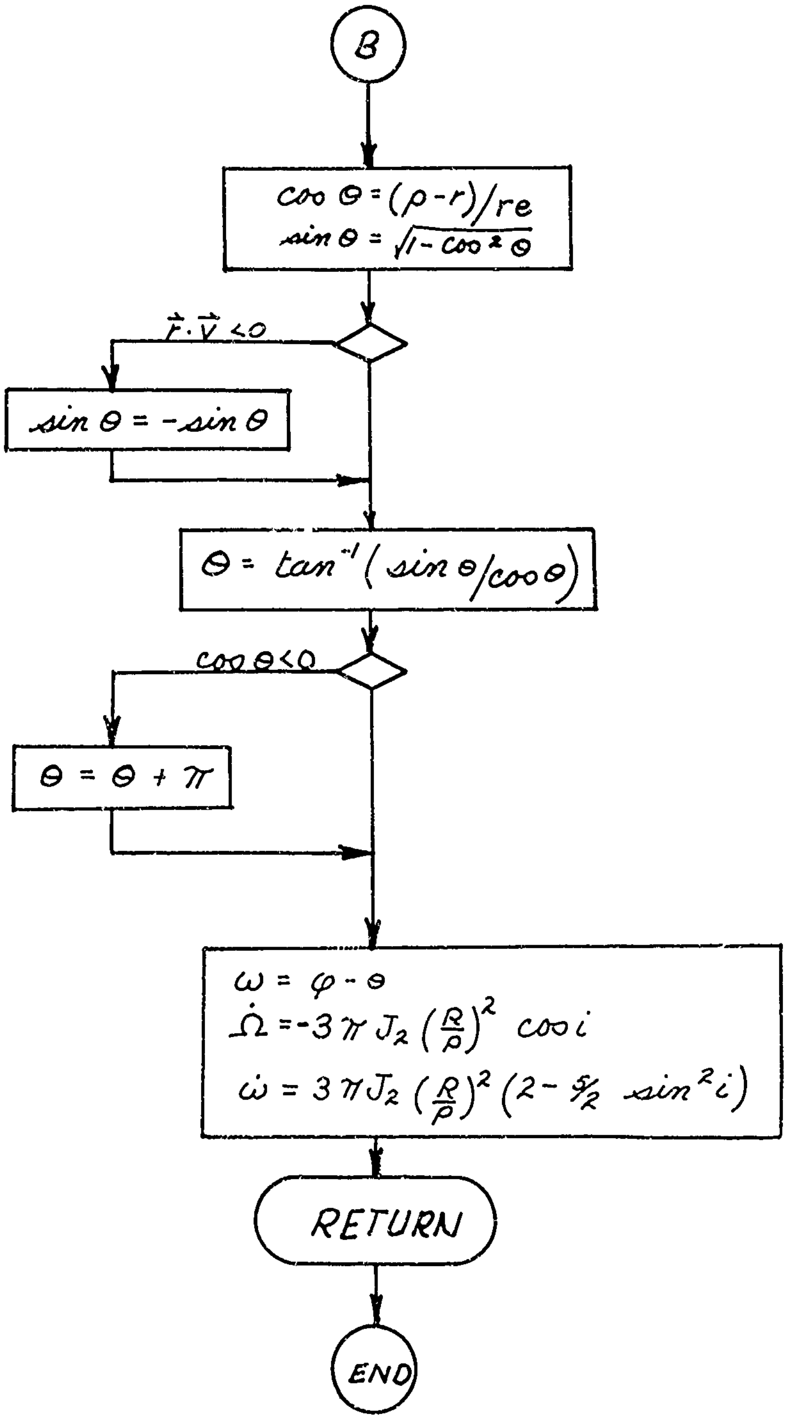
References

1) Townsend, G. E., "Orbital Mechanics" in the Orbital Flight Handbook, Volume 1, Part 1, Chapter 3 NASA - SP-33, (1963).

Computational Logic:




A
cos i < 0
i = i + π
sin Ω = h₁ / sin i
cos Ω = h₂ / sin i
Ω = tan⁻¹ (sin Ω / cos Ω
cos Ω < 0
Ω = Ω + π
p = |r⃗ x v⃗|² / μ
a = rμ / (2μ - r v²)
e = √(1 - p/a)
a < 0
a = -a
X̂ = cos Ω x̂ + sin Ω ŷ + 0 ẑ
cos φ = r⃗ · X / |r|
sin φ = √(1 - cos² φ)
r₃ < 0
sin φ = - sin φ
φ = tan⁻¹ sin φ / cos φ
cos φ < 0
φ = φ + π
B


B
$\cos\theta = (p - r)/re$
$\sin\theta = \sqrt{1-\cos^2\theta}$
$\vec{r}\cdot\vec{V} < 0$
$\sin\theta = -\sin\theta$
$\theta = \tan^{-1}(\sin\theta/\cos\theta)$
$\cos\theta < 0$
$\theta = \theta + \pi$
$\omega = \varphi - \theta$
$\dot{\Omega} = -3\pi J_2 \left(\frac{R}{p}\right)^2 \cos i$
$\dot{\omega} = 3\pi J_2 \left(\frac{R}{p}\right)^2 \left(2 - \frac{5}{2}\sin^2 i\right)$
RETURN
END

```
      SUBROUTINE ELEMEN(R,V,GM,REQ ,A,E,P,THETA,OMEGA,OMEGAW,           ELEM0020
     1                 OINC,OMDOT,WDOT)                                 ELEM0030
C     THIS ROUTINE COMPUTES THE CONIC ORBIT ELEMENTS IN THE CLASSIC     ELEM0040
C              FORM FOR THE PURPOSE OF PROVIDING ADDITIONAL DATA FROM   ELEM0050
C              THE DATA FILTER                                          ELEM0060
C     R     = CARTESIAN POSITION VECTOR IN DESIRED FRAME                ELEM0070
C     V     = CARTESIAN VELOCITY VECTOR IN DESIRED FRAME                ELEM0080
C     GM    = GRAVITATIONAL CONSTANT IN UNITS COMPATIBLE WITH R AND V   ELEM0090
C     REQ   = EQUITORIAL RADIUS OF CENTRAL BODY                         ELEM0100
C     A     = SEMIMAJOR AXIS                                            ELEM0110
C     E     = ECCENTRICITY                                              ELEM0120
C     P     = SEMI-LATUS RECTUM                                         ELEM0130
C     THETA = TRUE ANOMALY OF DATE                       (RAD)          ELEM0140
C     OMEGA = RIGHT ASCENSION OF THE ASCENDING NODE      (RAD)          ELEM0150
C     OMEGAW= ARGUMENT OF PERIAPSE                       (RAD)          ELEM0160
C     OINC  = ORBITAL INCLINATION                        (RAD)          ELEM0170
C     OMDOT = SECULAR RATE OF CHANGE IN OMEGA            (RAD/PERIOD)   ELEM0180
C     WDOT  = SECULAR RATE OF CHANGE IN OMEGAW           (RAD/PERIOD)   ELEM0190
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   ELEM0200
C                                                                       ELEM0210
      DIMENSION R(3),V(3),X(3),H(3)                                     ELEM0220
C                                                                       ELEM0230
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   ELEM0240
C                                                                       ELEM0250
C     COMPUTATION OF PLANAR ORIENTATION * * * * * * * * * * * * * * *   ELEM0260
      CALL CROSS(R,V,H)                                                 ELEM0270   2
      HMAG =AMAG(H)                                                     ELEM0280   4
      IF( HMAG -1.) 1,2,1                                               ELEM0290
    1 DO 3 I=1,3                                                        ELEM0300
      H(I) =H(I)/ HMAG                                                  ELEM0310
    3 CONTINUE                                                          ELEM0320
    2 CI =H(3)                                                          ELEM0330
      PI =3.1415926                                                     ELEM0340
      SI = SQRT ( 1. -CI*CI )                                           ELEM0350  16
      IF(CI) 5,6,5                                                      ELEM0360
    6 OINC =1.5707963                                                   ELEM0370
```

SID 65 1203-1

```
      ELEM         -  EFN   SOURCE STATEMENT  -  IFN(S)  -

        GO TO 11                                                       ELEM0380
      5 OINC = ATAN  (SI/CI)                                           ELEM0390    21
        IF(OINC) 21,18,21                                              ELEM0400
     18 OMEGA= 0.                                                      ELEM0410
        SOMEG = 0.                                                     ELEM0420
        COMEG = 1.                                                     ELEM0430
        GO TO 30                                                       ELEM0440
     21 IF (CI) 10,11,11                                               ELEM0450
     10 OINC = OINC +PI                                                ELEM0460
     11 SOMEG = H(1)/(SIN (OINC))                                      ELEM0470    29
        COMEG =-H(2)/(SIN (OINC))                                      ELEM0480    30
        IF(COMEG)12,13,12                                              ELEM0490
     13 OMEGA =PI/2.                                                   ELEM0500
        IF(SOMEG)14,15,15                                              ELEM0510
     14 OMEGA=1.5*PI                                                   ELEM0520
     15 GO TO 30                                                       ELEM0530
     12 OMEGA = ATAN ( SOMEG/COMEG )                                   ELEM0540    37
        IF( COMEG ) 20,30,30                                           ELEM0550
     20 OMEGA = OMEGA + PI                                             ELEM0560
C                                                                      ELEM0570
C       COMPUTATION OF A,E, AND P      * * * * * * * * * * * * * * * * ELEM0580
C                                                                      ELEM0590
     30 P = HMAG*HMAG /GM                                              ELEM0600
        RMAG = AMAG(R)                                                 ELEM0610    41
        DEN = -RMAG *DOT(  V) + 2.*GM                                  ELEM0620    42
        A = GM* RMAG /DEN                                              ELEM0630
        POA = P/A                                                      ELEM0640
        E = SQRT ( 1.-POA)                                             ELEM0650    43
        IF(A) 50,100,100                                               ELEM0660
     50 A=-A                                                           ELEM0670
C                                                                      ELEM0680
C       COMPUTATION OF THETA AND OMEGAW * * * * * * * * * * * * * * * * ELEM0690
C                                                                      ELEM0700
    100 X(1) = COMEG                                                   ELEM0710
        X(2) = SOMEG                                                   ELEM0720
        X(3) = 0.                                                      ELEM0730
        CPHI =DOT(R,X)/RMAG                                            ELEM0740    47
```

354

```
      ELEM        -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      IF(ABS(CPHI) -1.) 52,52,53                            ELEM0750
   53 SPHI=0.                                               ELEM0760
      GO TO 115                                             ELEM0770
   52 SPHI = SQRT ( 1.- CPHI*CPHI )                         ELEM0780   52
      IF(CPHI) 111,112,111                                  ELEM0790
  112 PHI =1.5707963                                        ELEM0800
      IF(R(3)) 113,250,250                                  ELEM0810
  113 PHI =4.7123889                                        ELEM0820
      GO TO 250                                             ELEM0830
  111 IF(R(3)) 99,115,115                                   ELEM0840
   99 SPHI =-SPHI                                           ELEM0850
  115 PHI = ATAN (SPHI/CPHI)                                ELEM0860   62
      IF(CPHI ) 200,250,250                                 ELEM0870
  200 PHI =PHI + 3.1415926                                  ELEM0880
  250 CT =(P-RMAG)/(RMAG *E )                               ELEM0890
      IF(ABS(CT)-1.) 62,62,63                               ELEM0900
   63 ST=0.                                                 ELEM0910
      GO TO 31                                              ELEM0920
   62 ST =SQRT (1.- CT*CT)                                  ELEM0930   70
      IF (DOT(R,V)) 36,59,59                                ELEM0940   71
   36 ST =-ST                                               ELEM0950
   59 IF(CT) 31,41,31                                       ELEM0960
   41 THETA =1.5707963                                      ELEM0970
      IF(ST) 42,33,33                                       ELEM0980
   42 THETA =4.7123889                                      ELEM0990
      GO TO 33                                              ELEM1000
   31 THETA = ATAN ( ST/CT)                                 ELEM1010   81
      IF(CT) 32,33,33                                       ELEM1020
   32 THETA = THETA + 3.1415926                             ELEM1030
   33 OMEGAW = PHI -THETA                                   ELEM1040
      IF(OMEGAW) 35,37,37                                   ELEM1050
   35 OMEGAW =6.2831852 + OMEGAW                            ELEM1060
C                                                           ELEM1070
C     COMPUTATION OF THE SECULAR PERTURBATION TERMS * * * * * * * * * * ELEM1080
   37 CONST = 3.*PI*.00108228*(REQ/P)**2                    ELEM1090
      OMDOT = -CONST*CI                                     ELEM1100
      WDOT  =  CONST*(2.-2.5*SI*SI)                         ELEM1110
```

SID 65 1203-1

```
C                                                              ELEM1120
C                                                              ELEM1130
      RETURN                                                   ELEM1140
      END                                                      ELEM1150
```

SUBROUTINE ELEMEN

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
X	00001	R	H	00004	R			

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
HMAG	00007	R	CI	00010	R	PI	00011	R
SI	00012	R	SOMEG	00013	R	COMEG	00014	R
RMAG	00015	R	JEN	00016	R	POA	00017	R
CPHI	00020	R	SPHI	00021	R	PHI	00022	R
CT	00023	R	ST	00024	R	CONST	00025	R

ENTRY POINTS

ELEMEN	SECTION	3

SUBROUTINES CALLED

CROSS	SECTION	4	AMAG	SECTION	5	SQRT	SECTION	6
ATAN	SECTION	7	SIN	SECTION	8	DOT	SECTION	9
SYSLOC	SECTION	10						

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1	6A	00065	2	15A	00073	3	12A	00071
5	20A	00116	6	18A	00113	11	28A	00143
21	25A	00135	18	23A	00130	30	40A	00217

STORAGE MAP

ELEM									
10	27A	00140	12	36A	00201	13	32A	00167	
14	34A	00175	15	35A	00200	20	39A	00214	
50	45A	00270	100	46A	00272	52	51A	00315	
53	49A	00313	115	61A	00346	111	58A	00341	
112	54A	00331	113	56A	00336	250	65A	00364	
99	60A	00344	200	64A	00361	62	69A	00404	
63	67A	00402	31	80A	00441	36	73A	00425	
59	74A	00427	41	76A	00431	42	78A	00436	
33	84A	00457	32	83A	00454	35	86A	00465	
37	87A	00470							

DECK LENGTH IN OCTAL IS 00641.

2.4.1.6 Subroutine DATAPE

Purpose: Reads a specially generated magnetic tape containing smoothed and ordered coordinate data.

Deck Name: DAPE

Calling Sequence: CALL DATAPE (KOUNT)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Description
0	TW TF	T_{DATA}	1 1	WRK(61) WRK(62)	The whole and fractional part of the number of days relative to 1950.0 at which the next observation will occur (1950.0 = J.D. 2433282.423)
0	ISTN	-	1	WRK(63)	The number for the observing station
0	ITYPE	-	1	WRK(64)	The type of data: 1, Range 2, Range-Rate 3, Azimuth, Elevation 4, Range, Range-Rate 5, Range, Azimuth, Elevation 6, Range-Rate, Azimuth, Elevation
0	ODATA	-	3	WRK(65)	The components of the observation vector
I/O	KOUNT	-	1	ARG	Control indicator. Code: 1, Do not return next point (INPUT) 2, Return next point (INPUT) 3, No more data points on tape (OUTPUT)

Subroutines Required:	None
Functions Required:	None
Approximate Storage Required:	650 (octal)
Restrictions:	Requires that the input data tape to be generated by FS4-305A be mounted on logical tape drive unit 9.

Nomenclature:

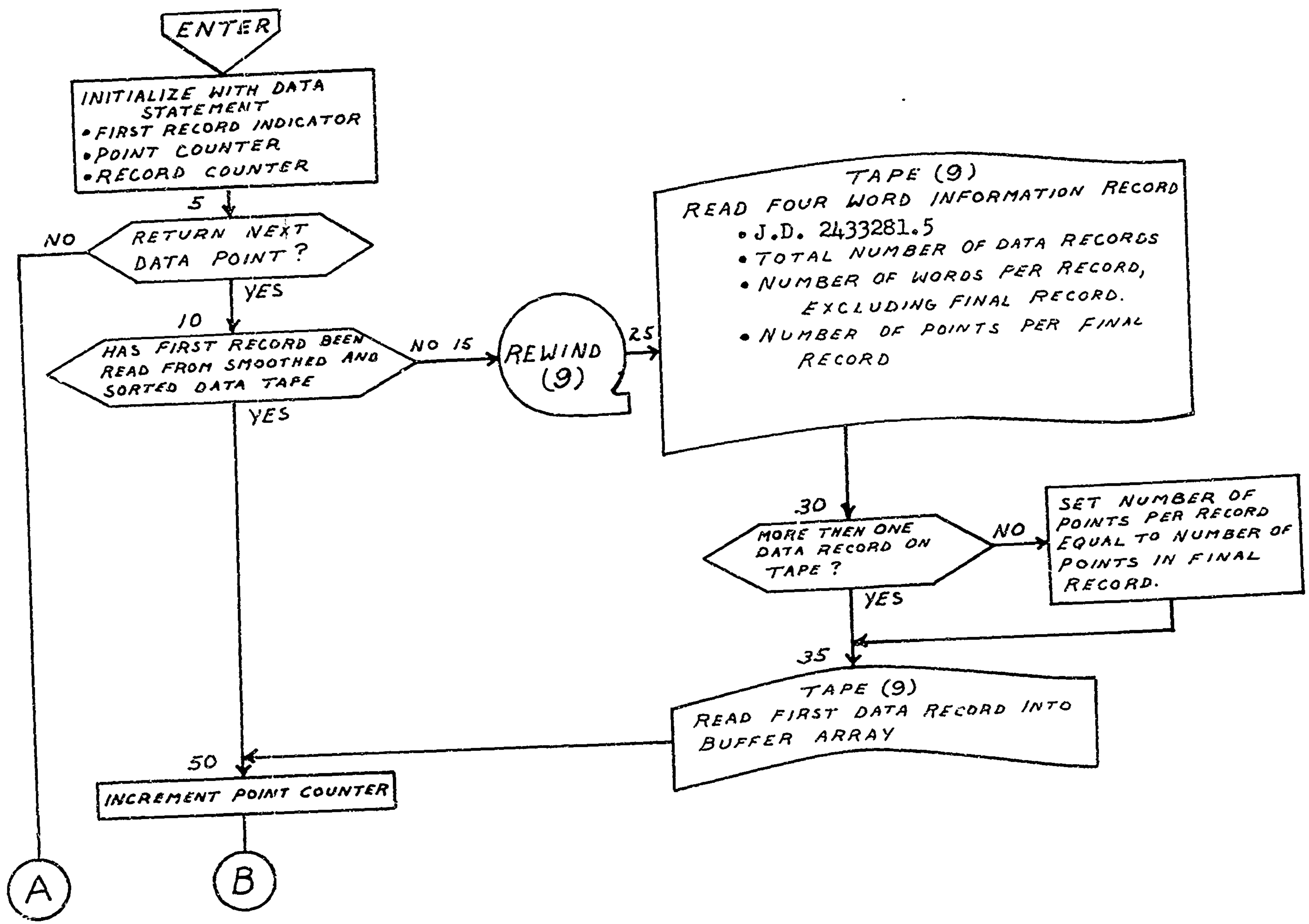
FØRTRAN Name	Dimensions	Description
A	7,37	Buffer array containing a single logical tape record.
IFIRST	----	First pass indicator
IGP	----	Logical record counter, IGP=1, NGPS
NGPS	----	Number of logical data records on tape.
NPERGP	----	Number of points per logical record, excluding final record.
NPREM	----	Number of points per final record.
NPØINT	----	Buffer array point index, NPØINT=1, (NPERGP or NPREM)
XJDREF	----	Program reference Julian date.**

Method:

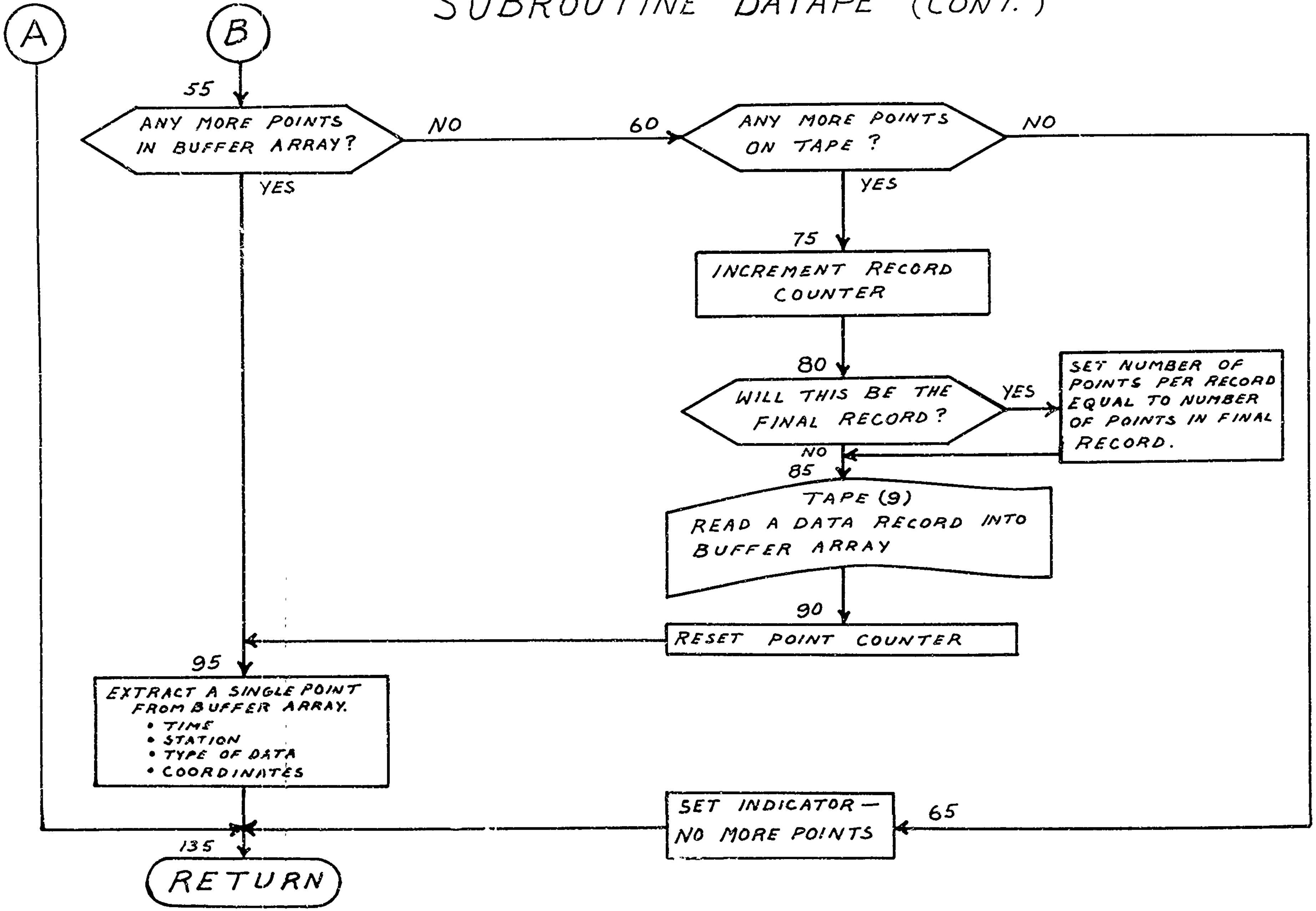
The first CALL statement to this subroutine mechanizes the input data tape prepared by the preliminary processor program. An information record containing J.D.2433281.5 (unused in this program), the number of logical records, and the number of data points* per record is read into the program. The first logical data record is then read into the buffer array A and, after extracting the first data point from the buffer, control is returned to the calling routine. Subsequent CALL statements extract single points sequentially from the buffer. After the final point within the buffer has been extracted, the next logical record is read in and the procedure is repeated until all data has been read from the tape at which time the count index (KOUNT) is set equal to 3 and control is returned to the norm program.

*A data point is defined as the ordered set TW, TF, ISTN, ITYPE, ØDATA.
**JD (24) 33282.423

SUBROUTINE DATAPE



SUBROUTINE DATAPE (CONT.)



```
C                                                                       DAPE0020
      SUBROUTINE DATAPE(KOUNT)                                          DAPE0030
C                                                                       DAPE0040
C*** FS4-305 ***            *** SUBROUTINE DATAPE ***                   DAPE0050
C                                                                       DAPE0060
C     PURPOSE,                                                          DAPE0070
C                                                                       DAPE0080
C         READS A SPECIALLY GENERATED MAGNETIC TAPE CONTAINING SMOOTHED DAPE0090
C              AND ORDERED OBSERVATION VECTORS AS FUNCTIONS OF TIME AS  DAPE0100
C              THE PRIMARY ARGUMENT                                     DAPE0110
C                                                                       DAPE0120
C     ARGUMENT DEFINITIONS,                                             DAPE0130
C                                                                       DAPE0140
C         TW    , TIME. INTEGER DAYS FROM 1950.0 (JD 2433282.423)       DAPE0150
C         TF    , TIME. FRACTIONAL DAY.    (U.T.)                       DAPE0160
C         ISTN , STATION FROM WHICH DATA WAS RECIEVED.                  DAPE0170
C         ITYPE, INDICATES TYPE OF DATA IN ODATA.                       DAPE0180
C                  1, RANGE                                             DAPE0190
C                  2, RANGE RATE                                        DAPE0200
C                  3, AZIMUTH,ELEVATION                                 DAPE0210
C                  4, RANGE,RANGE RATE                                  DAPE0220
C                  5, RANGE,AZIMUTH,ELEVATION                           DAPE0230
C                  6, RANGE RATE,AZIMUTH,ELEVATION                      DAPE0240
C         ODATA, COORDINATE DATA.                                       DAPE0250
C         KOUNT, CONTROL INDICATOR.                                     DAPE0260
C                  1, DO NOT RETURN NEXT POINT.    (INPUT)              DAPE0270
C                  2, RETURN NEXT POINT            (INPUT)              DAPE0280
C                  3, NO MORE DATA POINTS ON TAPE.(OUTPUT)              DAPE0290
C                                                                       DAPE0300
C     METHOD,                                                           DAPE0310
C                                                                       DAPE0320
C         LOGICAL DATA RECORDS ARE READ FROM TAPE INTO THE BUFFER       DAPE0330
C         ARRAY A. SINGLE DATA POINTS ARE THEN SEQUENTLY EXTRACTED      DAPE0340
C         FROM THE BUFFER BY EACH CALL OF DATAPE.                       DAPE0350
C                                                                       DAPE0360
C     RESTRICTIONS,                                                     DAPE0370
```

```
     DAPE        -  EFN   SOURCE STATEMENT  -  IFN(S)  -

C                                                                    DAPE0380
C         REQUIRES THAT THE INPUT DATA TAPE GENERATED BY FS4-305A BE DAPE0390
C         MOUNTED ON LOGICAL TAPE DRIVE UNIT 9.                      DAPE0400
C                                                                    DAPE0410
C                                                                    DAPE0420
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * DAPE0430
C                                                                    DAPE0440
      DIMENSION  CON(1), SAT(1), SDA(1), WRK(1), STT(1)              DAPE0450
C                                                                    DAPE0460
      COMMON DATA                                                    DAPE0470
C                                                                    DAPE0480
      EQUIVALENCE  (DATA(  1),CON), (DATA( 16),SAT), (DATA( 36),SDA) DAPE0490
     2             ,(DATA(286),STT), (DATA(391),WRK)                 DAPE0500
C                                                                    DAPE0510
      DIMENSION  ODATA(3)  ,A(7,37)                                  DAPE0520
C                                                                    DAPE0530
C                                                                    DAPE0540
      EQUIVALENCE            (WRK( 61),TW    ), (WRK( 62),TF     )   DAPE0550
     2  ,(WRK( 63),ISTN  ), (WRK( 64),ITYPE ), (WRK( 65),ODATA )     DAPE0560
C                                                                    DAPE0570
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * DAPE0580
C                                                                    DAPE0590
      DATA  IFIRST, NPOINT, IGP / 3*0 /                              DAPE0600
C                                                                    DAPE0610
C                                                                    DAPE0620
    5 IF(  KOUNT.EQ.1 ) GO TO 135                                    DAPE0630
   10 IF( IFIRST.NE.0 ) GO TO 50                                     DAPE0640
C                                 FIRST PASS, READ INFORMATION       DAPE0650
C                                 AND FIRST DATA RECORD.             DAPE0660
   15 REWIND 9                                                       DAPE0670    7
   20 IFIRST = 1                                                     DAPE0680
   25 READ (9)  XJDREF,NGPS,NPERGP,NPREM                             DAPE0690    9
   30 IF(NGPS.EQ.1)  NPERGP = NPREM                                  DAPE0700
   35 NEND = NPERGP                                                  DAPE0710
   40 READ (9) ((A(J,I),J=1,7),I=1,NEND)                             DAPE0720   17
   45 IGP = 1                                                        DAPE0730
C                                 INCREMENT BUFFER POINT COUNT.      DAPE0740
```

```
   50 NPØINT = NPØINT + 1                                                  DAPE0750
C                                       IF FINAL PØINT HAS BEEN EXTRACTEDDAPE0760
C                                       FRØM THE BUFFER, READ NEXT RECD. DAPE0770
   55 IF( NPØINT.LE.NEND ) GØ TØ 95                                        DAPE0780
C                                       READ NEXT LØGICAL RECØRD.  IF NØ DAPE0790
C                                       MØRE DATA, SET KØUNT EQUAL TØ 3  DAPE0800
C                                       AND RETURN.                      DAPE0810
   60 IF( IGP.NE.NGPS ) GØ TØ 75                                           DAPE0820
   65 KØUNT = 3                                                            DAPE0830
   70 GØ TØ 135                                                            DAPE0840
C                                                                        DAPE0850
   75 IGP = IGP + 1                                                        DAPE0860
   80 IF( IGP.EQ.NGPS ) NEND = NPREM                                       DAPE0870
   85 READ (9) ((A(J,I),J=1,7),I=1,NEND)                                   DAPE0880     41
   90 NPØINT = 1                                                           DAPE0890
C                                       EXTRACT PØINT FRØM BUFFER.       DAPE0900
   95 TW        = A(1,NPØINT)                                              DAPE0910
  100 TF        = A(2,NPØINT)                                              DAPE0920
  105 ISTN      = A(3,NPØINT)                                              DAPE0930
  110 ITYPE     = A(4,NPØINT)                                              DAPE0940
  115 ØDATA(1)  = A(5,NPØINT)                                              DAPE0950
  120 ØDATA(2)  = A(6,NPØINT)                                              DAPE0960
  125 ØDATA(3)  = A(7,NPØINT)                                              DAPE0970
  135 CONTINUE                                                             DAPE0980
  140 RETURN                                                               DAPE0990
      END                                                                  DAPE1000
```

STORAGE MAP

SUBROUTINE DATAPE

COMMON VARIABLES

COMMON BLOCK // ORIGIN 00001 LENGTH 00711

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
DATA	00000	R	DATA	00000	R	CON	00030	R
SAT	00017	R	SDA	00043	R	STT	00435	R
WRK	00606	R	IW	00702	R	TF	00703	R
ISTN	00704	I	ITYPE	00705	I	ODATA	00706	R

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
A	00712	R						

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
IFIRST	01315	I	XJDREF	01316	R	NGPS	01317	I
NPERGP	01320	I	NPREM	01321	I	NEND	01322	I
I	01323	I	IGP	01324	I	NPOINT	01325	I

ENTRY POINTS

DATAPE SECTION 4

SUBROUTINES CALLED

.FRWT.	SECTION	5	.FRDB.	SECTION	6	.UN09.	SECTION	7
.FRLR.	SECTION	8	.FBLT.	SECTION	9	.FBDT.	SECTION	10

STØRAGE MAP

CC.1	SECTIØN	11	CC.2	SECTIØN	12	CC.3	SECTIØN	13
CC.4	SECTIØN	14	SYSLØC	SECTIØN	15			

EFN IFN CØRRESPØNDENCE

EFN	IFN	LØCATIØN	EFN	IFN	LØCATIØN	EFN	IFN	LØCATIØN
5	1A	01340	135	66A	01542	10	4A	01344
50	28A	01433	15	7A	01350	20	8A	01354
25	9A	01356	30	13A	01373	35	16A	01400
40	17A	01402	45	27A	01431	55	29A	01436
95	52A	01513	60	32A	01443	75	37A	01452
65	35A	01447	70	36A	01451	80	38A	01455
85	41A	01462	90	51A	01511	100	54A	01520
105	56A	01522	110	58A	01527	115	60A	01534
120	62A	01536	125	64A	01540	140	67A	01542

DECK LENGTH IN ØCTAL IS 00655.

2.5 Matrix and Vector Algebra Group

The routines of this group are general purpose algebraic routines for vectors and matrices, and there are no inherent limitations in the formulations or codings which restrict application (though there are fixed dimensions in some cases which should be altered to provide expanded coverage). Accuracy is assured in those cases where single precision arithmetic may produce loss of significances by employing double precision.

2.5.1 Subroutine MATMPY (Matrix Multiplication)

Purpose: MATMPY is designed to multiply any two conformable single precision matrices (with less than 70 elements each) to obtain the single precision product. All operations interior to the routine are performed in double precision to control roundoff and loss of significance.

Deck Name: MXPY

Calling Sequence: CALL MATMPY (C, I, K, D, K, J, CD)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	C	C	I (rows) K (columns)	Arg	Array of numbers to be used in the premultiplication of D by C.
I	D	D	K (rows) J (columns)	Arg	Array of numbers to be premultiplied by the matrix C.
I	CD	CD	I (rows) J (columns)	Arg	Product array.

Subroutines Required: None

Functions Required: None

Approximate Deck Length: 700 (decimal)
1300 (octal)

```
      SUBROUTINE MATMPY(C,I,K,D,KK,J,CD )                               MATMPY02
      DIMENSION C(1), D(1), CD(1)                                       MATMPY04
      DOUBLE PRECISION A( 70),B( 70),AB( 70)                            MATMPY05
C                                                                       MATMPY06
C     THE MATRIX A HAS I ROWS AND K COLUMNS                             MATMPY08
C                                                                       MATMPY10
C     THE MATRIX B HAS K ROWS AND J COLUMNS                             MATMPY12
C                                                                       MATMPY14
C     THE PRODUCT AB HAS I ROWS AND J COLUMNS                           MATMPY16
C                                                                       MATMPY18
C     NA, NB, NAB  ARE THE MAXIMUM NUMBER OF ROWS GIVEN IN THE DIMENSIONMATMPY20
C                  STATEMENT IN THE MAIN PROGRAM FOR MATRICES A, B, AND MATMPY22
C                  AB, RESPECTIVELY                                     MATMPY24
C                                                                       MATMPY26
C                                                                       MATMPY27
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * MATMPY27
C                                                                       MATMPY27
      XSUBF(NROWS, I, J) = NROWS*(J-1) + I                              MATMPY28
C                                                                       MATMPY28
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * MATMPY28
C                                                                       MATMPY28
      DO 1 L =1,I                                                       MATMPY29
      DO 1 M =1,K                                                       MATMPY29
      NUMB = XSUBF(I,L,M)                                               MATMPY29
    1 A(NUMB) = C(NUMB)                                                 MATMPY29
      DO 2 L =1,K                                                       MATMPY29
      DO 2 M =1,J                                                       MATMPY29
      NUMB = XSUBF(K,L,M)                                               MATMPY29
    2 B(NUMB) = D(NUMB)                                                 MATMPY29
C                                                                       MATMPY29
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * MATMPY29
C                                                                       MATMPY29
      NA  = I                                                           MATMPY29
      NB  = K                                                           MATMPY29
      NAB = I                                                           MATMPY29
      DO  10  L=1, I                                                    MATMPY30
```

```
      DØ  10  M=1, J                                                    MATMPY32
      IAB = XSUBF(NAB, L, M)                                            MATMPY34
      AB(IAB) = 0.0                                                     MATMPY36
      DØ  10  N=1, K                                                    MATMPY38
      IA = XSUBF(NA, L, N)                                              MATMPY40
      IB = XSUBF(NB, N, M)                                              MATMPY42
C                                                                       MATMPY44
   10 AB(IAB)= AB(IAB) + A(IA)*B(IB)                                    MATMPY46
C                                                                       MATMPY47
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * MATMPY48
C                                                                       MATMPY49
      DØ 20 L =1,I                                                      MATMPY50
      DØ 20 M =1,J                                                      MATMPY51
      NUMB = XSUBF(I,L,M)                                               MATMPY52
   20 CD(NUMB) = AB(NUMB)                                               MATMPY53
      RETURN                                                            MATMPY54
      END                                                               MATMPY55
```

STØRAGE MAP

SUBRØUTINE MATMPY

DIMENSIØNED PRØGRAM VARIABLES

SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE
A	0C001	D	B	00215	D	AB	C0431	D

UNDIMENSIØNED PRØGRAM VARIABLES

SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE
L	0C645	I	M	00646	I	NUMB	C0647	I
NA	00650	I	NB	00651	I	NAB	00652	I
IAB	0C653	I	N	0C654	I	IA	00655	I
IB	C0656	I						

ENTRY PØINTS

MATMPY SECTIØN 3

SUBRØUTINES CALLED

E.1	SECTIØN	4	E.2	SECTIØN	5	E.3	SECTIØN	6
E.4	SECTIØN	7	CC.1	SECTIØN	8	CC.2	SECTIØN	9
CC.3	SECTIØN	10	CC.4	SECTIØN	11	SYSLØC	SECTIØN	12

EFN IFN CØRRESPØNDENCE

EFN	IFN	LØCATIØN	EFN	IFN	LØCATIØN	EFN	IFN	LØCATIØN
1	9A	00741	2	20A	01007	10	39A	01134
20	53A	01207						

DECK LENGTH IN ØCTAL IS 01307.

2.5.2 Subroutine MTINV (Matrix Inverse)

Purpose: MTINV computes the inverse of a nonsingular square array (of up to 36 elements) using the theorem which states that if a sequence of row operations will reduce a matrix to the identity matrix, then the same series of operations performed on the identity matrix will produce the inverse. (Error control is maintained internal to the routine with double precision arithmetic though input and output are single precision).

Deck Name: INV

Calling Sequence: CALL MTINV (B, ES, N)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	B	B	N X N	Arg	The N X N array of numbers to be inverted. (single precision)
O	ES	B^{-1}	N X N	Arg	The N X N inverse of B. (single precision)
I	N	N	1	Arg	The dimension of B

Subroutines Required: CHOOSE (Check for singular B)

Functions Required: None

Approximate Deck Length: 500 (decimal)
740 (octal)

```
      SUBROUTINE MTINV(B,ES,N )                                         10700010
C     COMPUTES THE INVERSE ,E, OF AN N BY N MATRIX ,B, USING THE        10700020
C     FOLLOWING THEOREM. IF A SEQUENCE OF ROW OPERATIONS WILL REDUCE A  10700030
C     MATRIX TO THE IDENTITY MATRIX, THEN THE SAME SEQUENCE OF          10700040
C     OPERATIONS WILL REDUCE THE IDENTITY MATRIX TO THE INVERSE.        10700050
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - 10700055
C     NOTE ** BOTH MATRICES MUST BE DIMENSIONED N BY N IN THE MAIN      10700058
C     PROGRAM .(THERE IS NO MAXIMUM VALUE FOR N ).                      10700060
      N=N                                                               10700065
      DIMENSION B(N,N),ES(N,N)                                          10700069
      DOUBLE PRECISION A(6,6),AMM,E(6,6),AIM,SCALE,SUM                  10700070
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - 10700071
C                            SCALE MATRIX TO REDUCE POSSIBLE LOSS OF    10700072
C                            SIGNIFICANCE AND EQUATE TO A               10700073
      SUM = 0.                                                          10700074
      DO 1 I=1,N                                                        10700075
      DO 1 J=1,N                                                        10700076
    1 SUM = SUM + B(I,J)                                                10700077
      X=N*N                                                             10700079
      SCALE = (SUM/ X)                                                  10700078
      DO 2 I=1,N                                                        10700080
      DO 2 J=1,N                                                        10700081
    2 A(I,J) = B(I,J) / SCALE                                           10700082
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - 10700083
C                            SET E EQUAL TO THE IDENTITY MATRIX         10700084
      DO 5 I=1,N                                                        10700090
      DO 5 J=1,N                                                        10700100
      IF (J-I) 6,7,6                                                    10700110
    7 E(I,J) = 1.D0                                                     10700120
      GO TO 5                                                           10700130
    6 E(I,J) = 0.D0                                                     10700140
    5 CONTINUE                                                          10700150
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - 10700155
      DO 80 M=1,N                                                       10700200
      CALL CHOOSE (A,E,M,N)                                             10700205
      AMM = A(M,M)                                                      10700210
```

42

```
      INV          -  EFN   SØURCE STATEMENT  -  IFN(S)  -

C                        REDUCE A(M,M) TØ 1 BY DIVIDING RØW M BY A(M,M).10700220
C                        PERFØRM THE SAME ØPERATIØN ØN E.               10700230
      DØ 20 J=1,N                                                        10700240
         A(M,J)   = A(M,J)/AMM                                           10700250
   20    E(M,J)   = E(M,J)/AMM                                           10700260
C-  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 10700265
C          REDUCE ALL ELEMENTS IN CØLUMN M, EXCEPT A(M,M), TØ ZERØ.      10700270
      DØ 72 I=1,N                                                        10700280
         IF (I-M) 25, 72, 25                                             10700290
   25    AIM      = A(I,M)                                               10700300
C                   SUBTRACT RØWS                                        10700310
      DØ 70 J=1,N                                                        10700320
               A(I,J)  = A(I,J) - AIM*A(M,J)                             10700330
   70          E(I,J)  = E(I,J) - AIM*E(M,J)                             10700340
   72    CØNTINUE                                                        10700350
   80 CØNTINUE                                                           10700360
C-  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 10700365
C                            RESCALE THE INVERSE AND CØNVERT TØ SINGLE   10700366
C                            PRECISIØN                                   10700367
      DØ 85 I=1,N                                                        10700368
      DØ 85 J=1,N                                                        10700369
   85 ES(I,J) = E(I,J)/SCALE                                             10700370
C-  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 10700371
      RETURN                                                             10700372
      END                                                                10700380
```

STORAGE MAP

SUBROUTINE MTINV

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
A	00001	D	E	00111	D			

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
AMM	00221	D	AIM	00223	D	SCALE	00225	D
SUM	00227	D	I	00231	I	J	00232	I
X	00233	R	M	00234	I			

ENTRY POINTS

MTINV	SECTION	3

SUBROUTINES CALLED

CHOOSE	SECTION	4	E.1	SECTION	5	E.2	SECTION	6
E.3	SECTION	7	E.4	SECTION	8	CC.1	SECTION	9
CC.2	SECTION	10	CC.3	SECTION	11	CC.4	SECTION	12
SYSLOC	SECTION	13						

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1	8A	00276	2	18A	00354	5	35A	00421
6	33A	00417	7	30A	00414	8C	70A	00570
20	49A	00470	72	68A	00565	25	56A	00530
70	63A	00554	85	78A	00621			

DECK LENGTH IN ØCTAL IS 00744.

2.5.2.1 Subroutine CHOOSE

Purpose: CHOOSE is utilized in conjunction with MTINV to determine if the matrix (of order less than 6) is sufficiently non-singular to allow the inverse to be constructed without excessive numerical difficulty.

Deck Name: CHSE

Calling Sequence: CALL CHOOSE (A, E, M, N)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	A	A	N X N	Arg	The array of numbers which is being reduced to the identity matrix.
I/O	E	E	N X N	Arg	The inverse being constructed from A.
I	M	-	1	Arg	A row counter for the operations being performed on A.
I	N	-	1	Arg	Order of the square array A.

Subroutine Required: None

Functions Required: None

Approximate Deck Length: 200 (decimal)
270 (octal)

```
     CHSE          -  EFN   SOURCE STATEMENT  -  IFN(S)  -

                SUBROUTINE CHOOSE(A,E,M,N)
C        IF THE DIAGONAL ELEMENT,A(M,M), OF THE MATRIX TO BE INVERTED IS   107A0000
C       ZERO, THE ROW WITH THE MAXIMAL ELEMENT IS CHOSEN AND INTERCHANGED   107A0010
C       WITH ROW M.                                                         107A0020
         N=N                                                                107A0030
         DOUBLE PRECISION A(6,6),E(6,6),EMAX,ABSEL,B                        107A0035
         IF (M - N ) 10 ,5 ,10                                              107A0040
       5 IF (ABS (A( N,N )) - .1D-30)40,40,70                               107A0050
      10   EMAX    =  ABS (A(M,M))                                          107A0060
           IROW    =  M                                                     107A0070
           M1      =  M + 1                                                 107A0080
           DO 30 I=  M1,N                                                   107A0090
             ABSEL=  ABS (A(I,M))                                           107A0100
             IF(EMAX-ABSEL) 20, 30, 30                                      107A0110
      20     EMAX =  ABSEL                                                  107A0120
             IROW =  I                                                      107A0130
      30     CONTINUE                                                       107A0140
           IF(EMAX - .1D-30)40, 50, 50                                      107A0150
      40   WRITE (6,45)                                                     107A0160     21
      45   FORMAT(28HOSINGULAR MATRIX, NO INVERSE)                          107A0170
         RETURN                                                             107A0180
      50 DO 60 I=1,N                                                        107A0190
             B            =  A(M,I)                                         107A0200
             A(M,I)       =  A(IROW,I)                                      107A0210
             A(IROW,I)    =  B                                              107A0220
             B            =  E(M,I)                                         107A0230
             E(M,I)       =  E(IROW,I)                                      107A0240
      60     E(IROW,I)    =  B                                              107A0250
      70 RETURN                                                             107A0260
         END                                                                107A0270
```

STØRAGE MAP

SUBRØUTINE CHØØSE

UNDIMENSIØNED PRØGRAM VARIABLES

SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE
EMAX	00001	D	ABSEL	00003	D	B	00005	D
IRØW	00007	I	M1	00010	I	I	00011	I

ENTRY PØINTS

CHØØSE	SECTIØN	3

SUBRØUTINES CALLED

.FWRD.	SECTIØN	4	.UN06.	SECTIØN	5	.FFIL.	SECTIØN	6
.FCNV.	SECTIØN	7	SYSLØC	SECTIØN	8			

EFN IFN CØRRESPØNDENCE

EFN	IFN	LØCATIØN	EFN	IFN	LØCATIØN	EFN	IFN	LØCATIØN
10	7A	00055	5	4A	00041	4C	21A	00136
70	37A	00215	30	17A	00127	2C	16A	00123
50	22A	00145	45	FØRMAT	00026	60	33A	00210

DECK LENGTH IN ØCTAL IS 00271.

SIE 65-1203-1
379

2.3.5 Subroutine ADDMAT

Purpose: ADDMAT is designed to add two arbitrary vectors or matrices of the same order

Deck Name: ADDM

Calling Sequence: CALL ADDMAT (A, I, J, B, C)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	A, B	A, B	I (rows) J (columns)	Arg	Two arrays of numbers which are to be added
O	C	C	I (rows) J (columns)	Arg	Matrix containing the sum of A and B

Subroutines Required: None

Functions Required: None

Approximate Deck Length: 80 (decimal)

```
      SUBROUTINE ADDMAT(A,I,J,B,C )                                     ADDM0010
C     THIS ROUTINE ADDS ADY TWO MATRICES OR VECTORS OF SIMILAR DIMENSIONADDM0020
C                                                                       ADDM0030
      DIMENSION  A(I,J),B(I,J),C(I,J)                                   ADDM0040
      DO 1 K=1,I                                                        ADDM0050
      DO 1 L=1,J                                                        ADDM0060
    1 C(K,L) = A(K,L) + B(K,L)                                          ADDM0070
      RETURN                                                            ADDM0080
      END                                                               ADDM0090
```

STORAGE MAP

SUBROUTINE ADDMAT

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
K	00001	I						

ENTRY POINTS

ADDMAT SECTION 3

SUBROUTINES CALLED

SYSLOC SECTION 4

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1	7A	00036						

DECK LENGTH IN OCTAL IS 00135.

2.5.4 Subroutine SUBMAT

Purpose: SUBMAT subtracts the I by J array of numbers B from another array (A) of the same order

Deck Name: SUBM

Calling Sequence: CALL SUBMAT (A, I, J, B, C)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	A, B	A, B	I (rows) J (columns)	Arg	A is an arbitrary array to be reduced by a second array of the same dimension (B)
O	C	C	I (rows) J (columns)	Arg	C = A - B

Subroutine Required: None

Functions Required: None

Approximate Deck Length: 80 (decimal)

```
      SUBROUTINE SUBMAT(A,I,J,B,C)                                      SUBM0010
C     THIS ROUTINE SUBTRACTS TWO MATRICES OF SIMILAR DIMENSION          SUBM0020
C                                                                       SUBM0030
      DIMENSION A(I,J),B(I,J),C(I,J)                                    SUBM0040
      DO 1 K=1,I                                                        SUBM0050
      DO 1 L=1,J                                                        SUBM0060
    1 C(K,L) = A(K,L) - B(K,L)                                          SUBM0070
      RETURN                                                            SUBM0080
      END                                                               SUBM0090
```

STØRAGE MAP

SUBRØUTINE SUBMAT

UNDIMENSIØNED PRØGRAM VARIABLES

SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE
K	00001	I						

ENTRY PØINTS

SUBMAT SECTIØN 3

SUBRØUTINES CALLED

SYSLØC SECTIØN 4

EFN IFN CØRRESPØNDENCE

EFN	IFN	LØCATIØN	EFN	IFN	LØCATIØN	EFN	IFN	LØCATIØN
1	7A	00036						

DECK LENGTH IN ØCTAL IS 00135.

2.5.5 Subroutine TRANSP (Matrix Transportation)

Purpose: TRANSP constructs the transpose of an arbitrary array of numbers

Deck Name: TRSP

Calling Sequence: CALL TRANSP (A, N, M, B)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	A	A	N (rows) M (columns)	Arg	Array to be transposed.
O	B	A^T	M (rows) N (columns)	Arg	The array containing the transpose.

Subroutines Required: None

Functions Required: None

Approximate Deck Length: 80 (decimal)
140 (octal)

TRSP - EFN SOURCE STATEMENT - IFN(S) -

```
      SUBROUTINE TRANSP(A,N,M,B)                                         TRAN0010
      DIMENSION A(N,M),B(M,N)                                            TRAN0020
      DO 1 I=1,N                                                         TRAN0030
      DO 1 J=1,M                                                         TRAN0040
    1 B(J,I) = A(I,J)                                                    TRAN0050
      RETURN                                                             TRAN0060
      END                                                                TRAN0070
```

STØRAGE MAP

SUBRØUTINE TRANSP

UNDIMENSIØNED PRØGRAM VARIABLES

SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE
I	00001	I						

ENTRY PØINTS

TRANSP SECTIØN 3

SUBRØUTINES CALLED

SYSLØC SECTIØN 4

EFN IFN CØRRESPØNDENCE

EFN	IFN	LØCATIØN	EFN	IFN	LØCATIØN	EFN	IFN	LØCATIØN
1	7A	00037						

DECK LENGTH IN ØCTAL IS 00137.

2.5.6 Subroutine CROSS

Purpose: CROSS constructs the vector product of two arbitrary vectors A and B (in the following order C = A X B)

Deck Name: CROSS

Calling Sequence: CALL CROSS (A, B, C)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	A B	A B	3 3	Arg	The two vectors to be multiplied in the sense A X B
O	C	C	3	Arg	The vector product

Subroutines Required: None

Functions Required: None

Approximate Deck Length: 70

```
    CRØS          -  EFN   SØURCE STATEMENT  -  IFN(S)  -

SUBRØUTINE CRØSS (A,B,C)                              CRØSS001
DIMENSIØN A(3),B(3),C(3)                              CRØSS002
C(1)= A(2)*B(3)-A(3)*B(2)                             CRØSS003
C(2)= A(3)*B(1)-A(1)*B(3)                             CRØSS004
C(3)= A(1)*B(2)-A(2)*B(1)                             CRØSS005
RETURN                                                CRØSS006
END                                                   CRØSS007
```

```
                                   STØRAGE  MAP

                           SUBRØUTINE  CRØSS

                                   ENTRY PØINTS

 CRØSS       SECTIØN     3

                                   SUBRØUTINES CALLED

 SYSLØC      SECTIØN     4

                               EFN   IFN   CØRRESPØNDENCE
EFN        IFN        LØCATIØN   EFN        IFN        LØCATIØN   EFN        IFN        LØCATIØN
  DECK LENGTH IN ØCTAL IS     C0105.
```

2.5.7 Subroutine DOT

Purpose: DOT computes the scalar product of two 3-vectors

Deck Name: DOT

Calling Sequence: DOT (A, B)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	A, B	A, B	3, 3	Arg	The two 3-vectors to be utilized in constructing the scalar product.
O	DOT	A, B	-	-	The scalar product.

Subroutines Required: None

Functions Required: None

Approximate Deck Length: 50 (decimal)

```
    DØTMM        -  EFN   SØURCE STATEMENT  -  IFN(S)  -

FUNCTION DØT(A,B)                                         DØTC0001
DIMENSIØN A(3),B(3)                                       DØTC0002
DØT= A(1)*B(1)+ A(2)*B(2)+ A(3)*B(3)                      DØT00003
RETURN                                                    DØT00004
END                                                       DØT00005
```

```
DØTMM
                              STØRAGE  MAP

                      FUNCTIØN    DØT         TYPE     R

                         UNDIMENSIØNED PRØGRAM VARIABLES

SYMBØL     LØCATIØN    TYPE    SYMBØL    LØCATIØN    TYPE    SYMBØL    LØCATIØN    TYPE
F.000C      00001       R

                              ENTRY PØINTS

    DØT        SECTIØN    3

                              SUBRØUTINES CALLED

    SYSLØC     SECTIØN    4

                         EFN   IFN   CØRRESPØNDENCE

EFN      IFN      LØCATIØN   EFN      IFN      LØCATIØN   EFN      IFN      LØCATIØN
 DECK LENGTH IN ØCTAL IS    00061.
```

2.5.8 Subroutine AMAG

Purpose: AMAG constructs the scalar length of a vector

Deck Name: AMAG

Calling Sequence: AMAG (A)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	A	A	3	Arg	The 3-vector (in cartesian coordinates) for which the length is desired.
O	AMAG	A	1	-	Vector magnitude.

Subroutines Required: None

Functions Required: DOT (Scalar product)

Approximate Deck Length: 30 (decimal)

```
    AMAGM        -  EFN   SØURCE STATEMENT  -  IFN(S)  -

FUNCTIØN AMAG(A)                                       MAG00001
DIMENSIØN A(3)                                         MAG00002
AMAG = SQRT (DØT(A,A))                                 MAG00003    2     3
RETURN                                                 MAG00004
END                                                    MAG00005
```

STORAGE MAP

FUNCTION AMAG TYPE R

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
F.000C	00001	R						

ENTRY POINTS

AMAG SECTION 3

SUBROUTINES CALLED

SORT SECTION 4 DOT SECTION 5 SYSLOC SECTION 6

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION

DECK LENGTH IN OCTAL IS 00043.

2.6 General Purpose Math Group

In addition to the functions and routines required for algebraic manipulation of vectors and matrices (as presented in the preceding discussions), there are several mathematical functions which are utilized in conjunction with the conic motion formulation (and the related state transition formulation) and in conjunction with the definition of azimuth of the vehicle relative to the tracking station which must be discussed. These routines, due to their general nature and the fact that they are employed at several points in the program, do not logically fall in one of the other main groups. They have, thus, been presented in a separate section on the following pages (no formulation or computational logic due to their simple form).

2.6.1 Subroutine SINH

Purpose: SINH computes the hyperbolic sine of a single precisio. argument expressed in radians.

Deck Name: SINH

Calling Sequence: SINH (X)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	X	X	1	Arg	Single precision argument (expressed in radians) for which SINH is to be evaluated.
O	SINH	sinh	1	-	Hyperbolic sine.

Subroutines Required: None

Functions Required: None

Approximate Deck Length: 100 (decimal)

```
     SINHM        -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      FUNCTION  SINH(XIN)                                                SINH0010
      X = XIN                                                            SINH0020
      IF(ABS (X)  -  88.028)10,10,200                                    SINH0030
   10 IF(ABS (X) - .34657359)20,20,40                                    SINH0040
   20 SINH  = (X ** 3)/6.  +  (X ** 5)/120.0  +  (X ** 7)/ 5040. + X     SINH0050
     1+ (X**9) / 362880.                                                 SINH0051    6
   30 RETURN                                                             SINH0060
   40 SINH  =  .5    * ( EXP (X) - EXP (-X))                             SINH0070    9     10
      GO TO 30                                                           SINH0080
  200 X  =  AMOD ( X,88.028)                                             SINH0090
      GO TO 10                                                           SINH0100
      END                                                                SINH0110
```

STØRAGE MAP

FUNCTIØN SINH TYPE R

UNDIMENSIØNED PRØGRAM VARIABLES

SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE	SYMBØL	LØCATIØN	TYPE
F.0000	CC001	R	X	00002	R			

ENTRY PØINTS

SINH	SECTIØN	3

SUBRØUTINES CALLED

.XP2.	SECTIØN	4	EXP	SECTIØN	5	CC.1	SECTIØN	6
CC.2	SECTIØN	7	CC.3	SECTIØN	8	CC.4	SECTIØN	9
SYSLØC	SECTIØN	10						

EFN IFN CØRRESPØNDENCE

EFN	IFN	LØCATIØN	EFN	IFN	LØCATIØN	EFN	IFN	LØCATIØN
10	3A	00037	200	12A	00132	20	5A	C0044
40	8A	00111	30	7A	00107			

DECK LENGTH IN ØCTAL IS CC165.

2.6.2 Subroutine COSH

Purpose: COSH computes the hyperbolic cosine of a single precision argument expressed in radians

Deck Name: COSH

Calling Sequence: COSH (X)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	X	X	1	Arg	Single precision argument (expressed in radians) for which COSH is to be evaluated.
O	COSH	cosh	1	-	Hyperbolic Cosine.

Subroutines Required: None

Functions Required: None

Approximate Deck Length: 100 (decimal)

```
       COSHM        -  EFN   SOURCE STATEMENT  -  IFN(S)  -

     FUNCTION COSH(XIN)                                              COSH0010
     X =  XIN                                                        COSH0020
     IF(ABS (X) - 88.028) 10,10,200                                  COSH0030
  10 IF(ABS (X) - .34657359)20,20,40                                 COSH0040
  20 COSH = 1. + (X ** 2)/ 2. + (X ** 4)/24. + (X ** 6) / 720.       COSH0050
    1+ (X**8) / 40320.                                               COSH0051    6
  30 RETURN                                                          COSH0060
  40 COSH    =  .5  * (EXP (X) + EXP (-X))                           COSH0070    9     10
     GO TO 30                                                        COSH0080
 200 X  =  AMOD (X,88.028)                                           COSH0090
     GO TO 10                                                        COSH0100
     END                                                             COSH0110
```

2.6.2 Subroutine COSH

Purpose: COSH computes the hyperbolic cosine of a single precision argument expressed in radians

Deck Name: COSH

Calling Sequence: COSH (X)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	X	X	1	Arg	Single precision argument (expressed in radians) for which COSH is to be evaluated.
0	COSH	cosh	1	-	Hyperbolic Cosine.

Subroutines Required: None

Functions Required: None

Approximate Deck Length: 100 (decimal)

```
     COSHM        -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      FUNCTION COSH(XIN)                                      COSH0010
      X =  XIN                                                COSHCC20
      IF(ABS (X) - 88.028) 10,1C,200                          COSHOC30
   10 IF(ABS (X) - .34657359)2C,20,40                         COSH0C40
   20 COSH = 1. + (X ** 2)/ 2. + (X ** 4)/24. + (X ** 6) / 720.  COSH0G50
     1+ (X**8) / 4032C.                                       COSH0051    6
   30 RETURN                                                  COSHOC60
   4C COSH    =  .5  * (EXP (X) + EXP (-X))                   COSH007C    9    10
      GO TO 30                                                COSH^080
  20C X  =  AMOD (X,88.C28)                                   COSH0G90
      GO TO 10                                                COSH01C0
      END                                                     COSH0110
```

COSHM

STORAGE MAP

FUNCTION COSH TYPE R

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
F.0000	00001	R	X	00002	R			

ENTRY POINTS

COSH	SECTION	3

SUBROUTINES CALLED

.XP2.	SECTION	4	EXP	SECTION	5	CC.1	SECTION	6
CC.2	SECTION	7	CC.3	SECTION	8	CC.4	SECTION	9
SYSLOC	SECTION	10						

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
10	3A	00037	200	12A	00124	20	5A	00044
40	8A	00104	30	7A	00102			

DECK LENGTH IN OCTAL IS 00157.

2.6.3 Subroutine ARKTNS

Purpose: ARKTNS computes the single precision arc tangent (when given the sine and cosine of the angle) and assigns the angle to the proper quadrant ($-\pi < \theta < \pi$ or $0 < \theta < 2\pi$)

Deck Name: ATAM

Calling Sequence: ARKTNS (N, X, Y)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	N	-	1	Arg	Fixed point number to identify the range of the are tangent N = 180 $-180 \leq \theta < 180$ N = 360 $0 \leq \theta < 360$
I	X	$\cos\theta$	1	Arg	Cosine of angle
I	Y	$\sin\theta$	1	Arg	Sine of angle
O	ARKTNS	θ	1	-	Arc tangent (Y/X)

Subroutine Required: None

Functions Required: ATAN (arc tangent function)

Approximate Deck Length: 160 (decimal)

```
    ****
        ATAM          -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      FUNCTION ARKTNS(N,X,Y)                                     ATANS030
      P=3.14159265                                               ATANS050
      K=N/180                                                    ATANS060
      IF(X) 30,20,30                                             ATANS062
   20 ARKTNS = P/2.                                              ATANS064
      GO TO 31                                                   ATANS066
   30 ARKTNS = ATAN (ABS (Y/X))                                  ATANS070     7
   31 IF(Y)1,2,3                                                 ATANS080
    1 GO TO (4,5),K                                              ATANS090
    4 IF (X)10,11,12                                             ATANS100
   10 ARKTNS = ARKTNS-P                                          ATANS110
      GO TO 9                                                    ATANS120
   11 ARKTNS = -P*.5                                             ATANS130
      GO TO 9                                                    ATANS140
   12 ARKTNS = -ARKTNS                                           ATANS150
      GO TO 9                                                    ATANS160
    5 IF (X)13,14,15                                             ATANS170
   13 ARKTNS = P+ARKTNS                                          ATANS180
      GO TO 9                                                    ATANS190
   14 ARKTNS = 3.*.5*P                                           ATANS200
      GO TO 9                                                    ATANS210
   15 ARKTNS = 2.*P-ARKTNS                                       ATANS220
      GO TO 9                                                    ATANS230
    2 IF (X)16,17,9                                              ATANS240
   16 ARKTNS = P                                                 ATANS250
      GO TO 9                                                    ATANS260
   17 WRITE(   6,18)                                             ATANS270    31
   18 FORMAT (90H0                     THE ARKTAN FUNCTION HAS   ATANS280
     1FAILED  -  ARKTAN(0/0) IS UNDEFINED/1H0)                   ATANS290
      GO TO 9                                                    ATANS300
    3 IF(X)8,19,9                                                ATANS310
    8 ARKTNS = P-ARKTNS                                          ATANS320
      GO TO 9                                                    ATANS330
   19 ARKTNS = .5*P                                              ATANS340
    9 RETURN                                                     ATANS350
      END                                                        ATANS360
```

```
****
   ATAM                                          STORAGE  MAP

                                    FUNCTION    ARKTNS      TYPE     R

                                        UNDIMENSIONED PROGRAM VARIABLES

SYMBOL       LOCATION      TYPE      SYMBOL       LOCATION      TYPE      SYMBOL       LOCATION      TYPE
F.0000        00001          R       P             00002          R       K             00003          I

                                              ENTRY POINTS

     ARKTNS       SECTION      3

                                              SUBROUTINES CALLED

     ATAN         SECTION      4        .FWRD.      SECTION      5        .FXEM.      SECTION      6
     .UN06.       SECTION      7        .FFIL.      SECTION      8        .FCNV.      SECTION      9
     SYSLOC       SECTION      10

                                        EFN   IFN   CORRESPONDENCE

EFN          IFN          LOCATION     EFN          IFN          LOCATION     EFN          IFN          LOCATION
30            6A           00057       20            4A           00053       31            8A           00072
1            10A           00075       2            27A           00145       3            33A           00162
4            11A           00104       5            19A           00123       10           13A           00107
11           15A           00113       12           17A           00120       9            38A           00174
13           21A           00126       14           23A           00132       15           25A           00140
16           29A           00150       17           31A           00153       18           FORMAT        00021
8            35A           00165       19           37A           00171
  DECK LENGTH IN OCTAL IS      00233.
```

2.6.4 Subroutine DERAQ

Purpose: DERAQ is intended to represent the Kronecker delta (i.e., δ_{ij}) and as such has the value of 1.0 or 0.0 depending on the arguments I and J

Deck Name: DERQ

Calling Sequence: DERAQ (I, J)

Input/Output:

I/O	FORTRAN Name	Math Name	Dimension	Common/ Argument	Definition
I	I, J	i, j	1, 1	Arg	Two fixed point variables defining the value of.
0	DERAQ		1	-	Kronecker delta.

Subroutine Required: None

Functions Required: None

Approximate Deck Length: 30 (decimal)

```
      FUNCTION DERAQ(I,J)                                          DRAQ0010
C                                                                  DRAQ0020
C     DERAQ IS THE KRONECKER DELTA FUNCTION                        DRAQ0030
C              = 1,    I = J                                       DRAQ0040
C              = 0,    I NOT EQUAL TO J                            DRAQ0050
C                                                                  DRAQ0060
      DERAQ = 0.                                                   DRAQ0070
      IF(I-J) 1,2,1                                                DRAQ0080
    2 DERAQ = 1.                                                   DRAQ0090
    1 RETURN                                                       DRAQC100
      END                                                          DRAQ0110
```

DRAQ

STORAGE MAP

FUNCTION DERAQ TYPE R

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE
F.0000	00001	R						

ENTRY POINTS

DERAQ SECTION 3

SUBROUTINES CALLED

SYSLOC SECTION 4

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
1	4A	00015	2	3A	00013			

DECK LENGTH IN OCTAL IS 00037.

3.0 Accuracy Tests

The program logic has been checked numerically against three precision programs.

1) AP110 - A general purpose trajectory program prepared for the Apollo program and employing an Encke formulation and an Adams integration logic.

2) SPACE - A JPL single precision cowell trajectory program employing a 4th order Runge-Kutta integration logic.

3) ITEM - A Goddard trajectory program employing an Encke formulation.

The results of these tests indicate that agreement can always be obtained through the sixth digit (frequently through the seventh) for integrations covering several orbits of the earth providing that the constants of the programs are compatible. This level of agreement is considered as a proof of the program logic. This conclusion is strengthened when it is observed that the basic logics of the three programs are somewhat different and that in no case could the agreement be expected through the eighth digit due to the restricted word length employed in single precision on the IBM 7094.

4.0 SAMPLE PROBLEM

4.1 SAMPLE PROBLEM DISCUSSION

Punched paper tapes with tracking data recorded by the Floyd Terminal for passes 6069 and 6070 of ECHO II were provided for the purpose of demonstrating the operation and accuracy of the differential corrections program described on the previous pages. Thus, the discussions which follow will enumerate the procedures involved in construction and solution of this sample problem.

4.2 ORBIT PARAMETERS

Passes 6069 and 6070 of ECHO II occurred on 27 April 1965. Thus, published data for this satellite was searched to determine an estimate of the position and velocity at some epoch prior to this date. This procedure was required to assure that the trajectory could be generated accurately so that the differential corrections process can converge to the desired trajectory.

The Goddard Space Flight Center on 30 April 1965 issued the following information pertaining to 1964 04-A (ECHO II) in TWX NR 054/301-474-4911 (the true equator of data frame of reference).

Epoch 0^h0^s UT, 20 April 1965

semimajor axis	7528.31 Km
eccentricity	0.02447
inclination	81.450°
Mean Anomaly	113.092°
argument of periapse	34.156°
right ascension of ascending mode	31.202°

Accordingly, this information was resolved into the corresponding position and velocity vectors. (Using subroutine POSVEL) the result was:

Epoch 0^h0^s UT, 20 April 1965

Radius Vector = -5915.9438 - 2918.1582 3783.0742 (Km)
Velocity Vector = -2.7444510 - 2.7299969 - 6.0748353 (Km/sec)

At this point, this information was updated to an epoch just prior to 6069 by employing the general perturbations program described in SID 1203-3. (The elements themselves could have been updated employing only the secular changes in ω and Ω. However, this approach was discarded to avoid introducing the slight errors associated with such a process.) The results of this process are:

Epoch 27.63865740, April 1965

Radius Vector = 4952.3943 1406.9609 -5362.9226 Km
Velocity Vector = 4.4573218 2.9062537 5.0928345 Km/sec

4.3 REDUCTION OF THE RAW DATA

4.3.1 RAW DATA TAPE

Input data for the sample problem consists of two paper tapes recorded by the Floyd tracking facility containing the following observations for the ECHO II satellite passes 6069 and 6070: elevation and azimuth in degrees, doppler reading in kilo-cycles, a lock-on indicator, and universal time in seconds. Since the IBM 7094 computing system utilized does not have a paper tape input capability, a short IBM 1401 program was written to transfer the data directly to magnetic tape with the format shown in Figure 1 (one physical file per pass). The leading information record for each data file contains the following indicators.

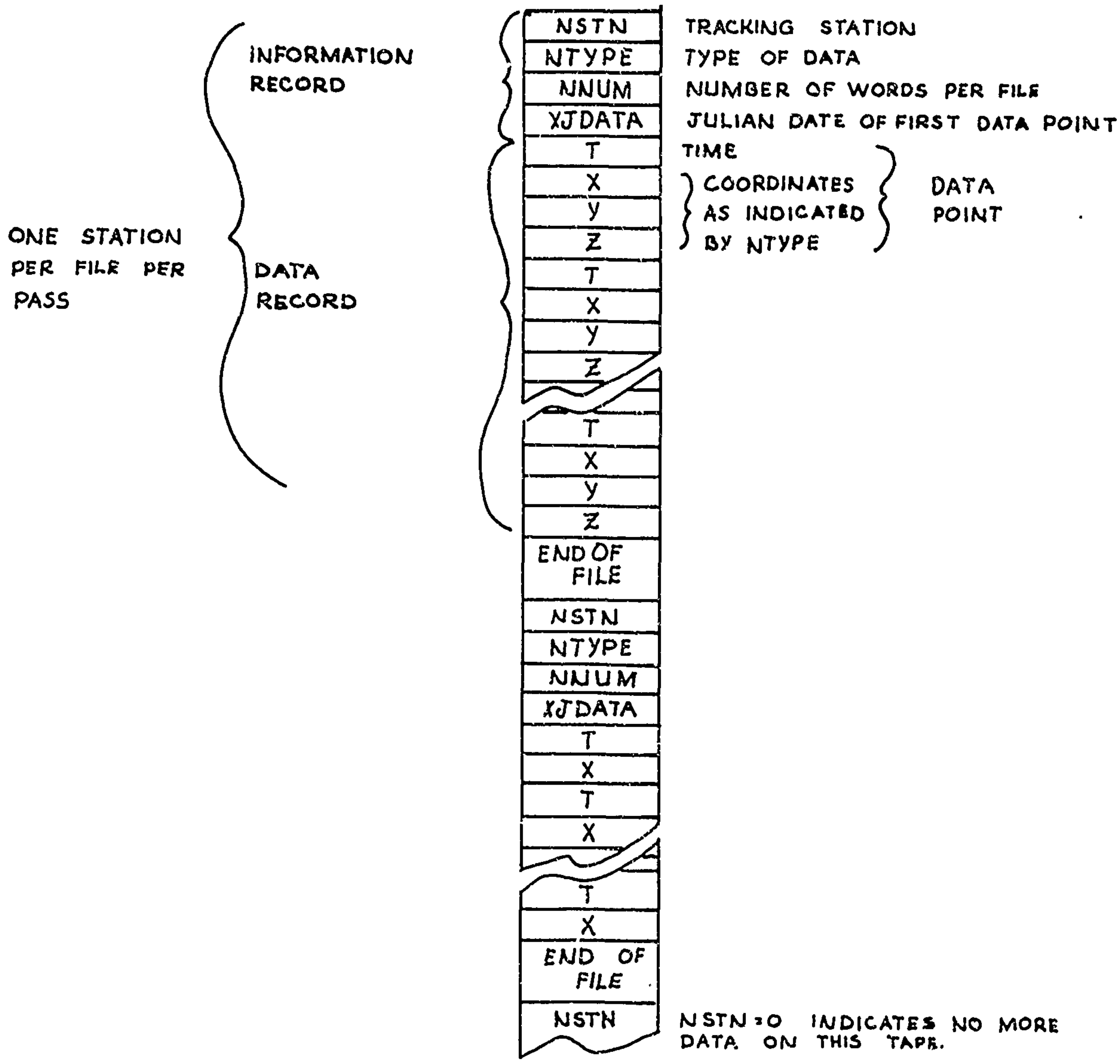
SYMBOL	DESCRIPTION	PASS 6069	PASS 6070
NSTN	Code indicating the tracking station recording the data (Floyd, see input)	1	1
NTYPE	Code Indicating the type of observed data (range rate, azimuth, elevation)	6	6
NNUM	Total number of words per file. (This was determined by the 1401 program in transcribing the data from punched paper tape)	3564	3392
XJDATA	Julian date (zero hour UT) of the first observation for each file. (27 April, 1965)	(24)38877.5	(24) 38877.5

A third file containing NSTN equal to zero indicates to the program that all data has been read from the tape.

Graphical representation of the raw data is presented in Figure 2 through 7 (the broad line apparent in these figures is in reality a series of points or plotting symbols). Examination of Figure 6 will disclose random irregularities of azimuth observations being recorded 360 degrees out of phase, especially prevalent for angles approaching 360 degrees (e.g., +354° recorded as -6°). These points were adjusted in the preliminary processor immediately prior to smoothing the data segment to assure that a consistent convention is employed in recording the observed and computed values of the data.

FIGURE 1

INPUT TAPE FORMAT
RAW DATA.



- FORTRAN IV
- BINARY MODE
- MULTI-PHYSICAL FILES
- LOGICALLY PACKED TIME AND COORDINATE DATA

This figure also serves to illustrate a data format which is not generally acceptable to the program. Although most of the data are observed to be sequenced chronologically, a few recording errors of the time are evident. At this time no attempt has been made to determine if such errors exist. Rather, it is considered to be the user's responsibility to assure that such errors have been eliminated.

Fortunately for this problem these errors do not affect the solution due to the manner employed in smoothing the raw data over short intervals of time. However, this circumstance exists only because the erroneous time values did not occur at the midpoint of a 20 consecutive point strip of data (see subroutine FIT in the preprocessor SID 65-1203-2). Since such fortuitous arrangement of errors cannot be guaranteed, remedial steps should be taken in recording the data.

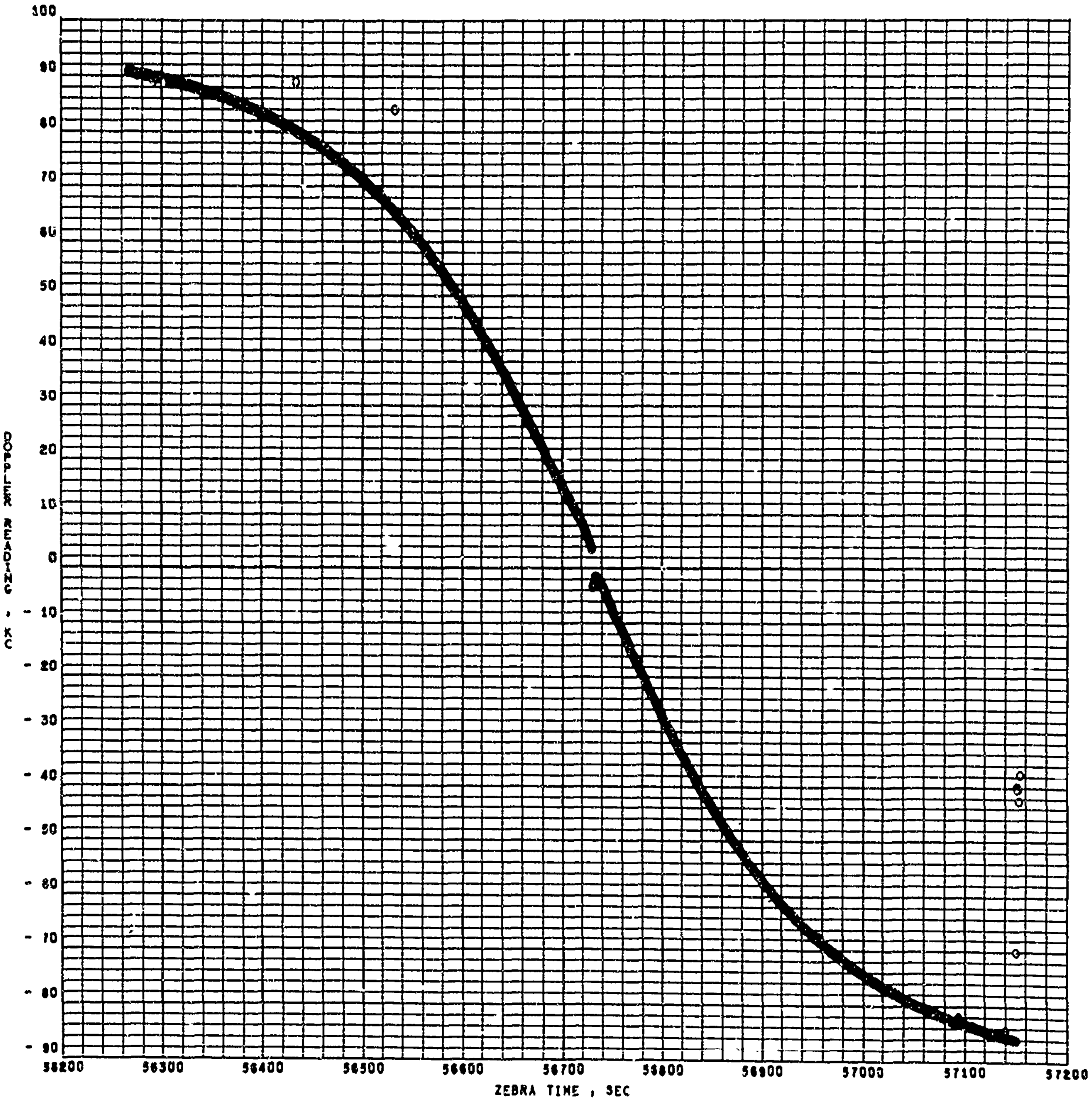


Figure 2 FIRST RAW DATA FILE
DOPPLER READING VS TIME
(Echo II, pass 6069)

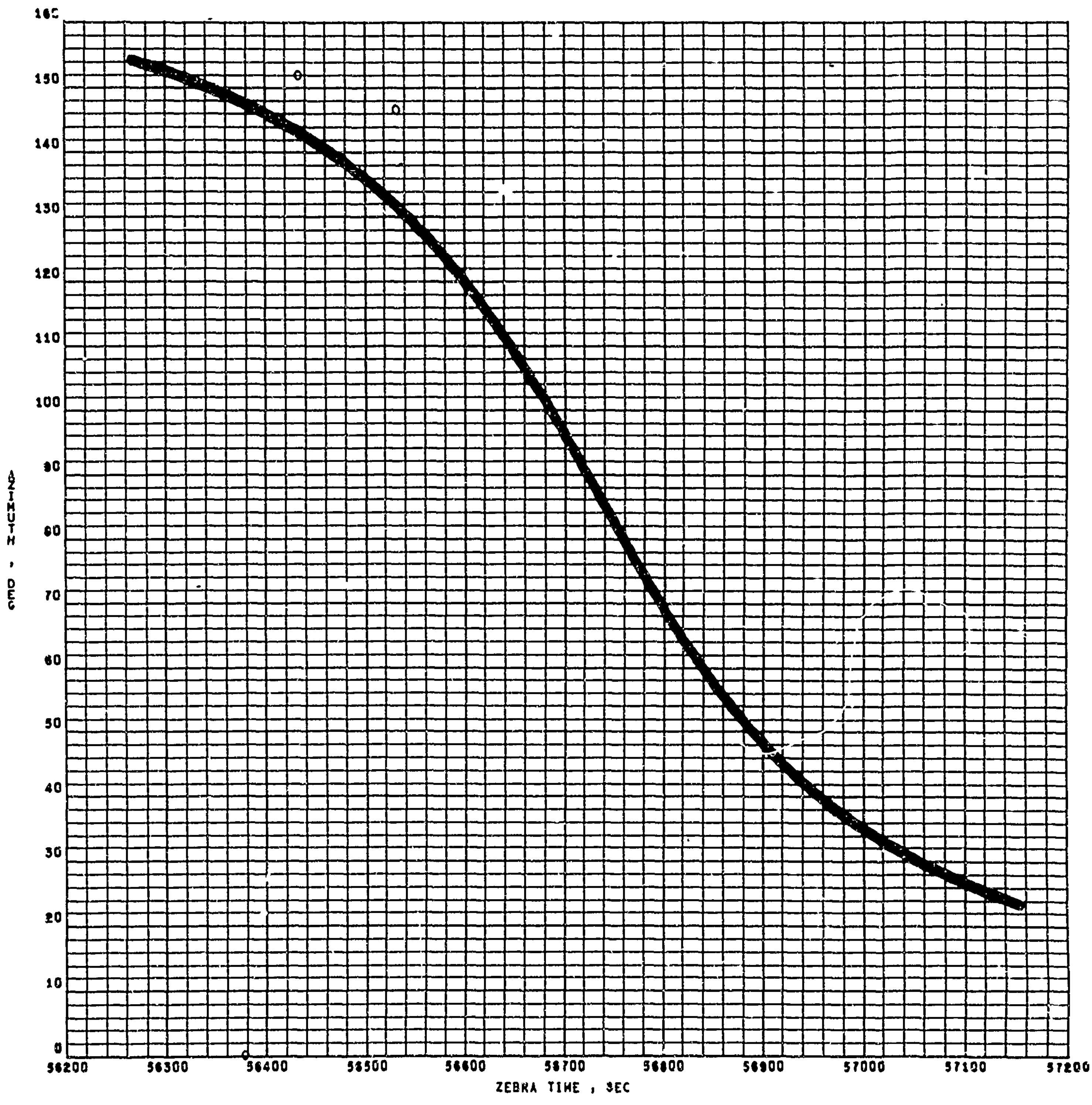


Figure 3 FIRST RAW DATA FILE
AZIMUTH VERSUS TIME
(Echo II, pass 6069)

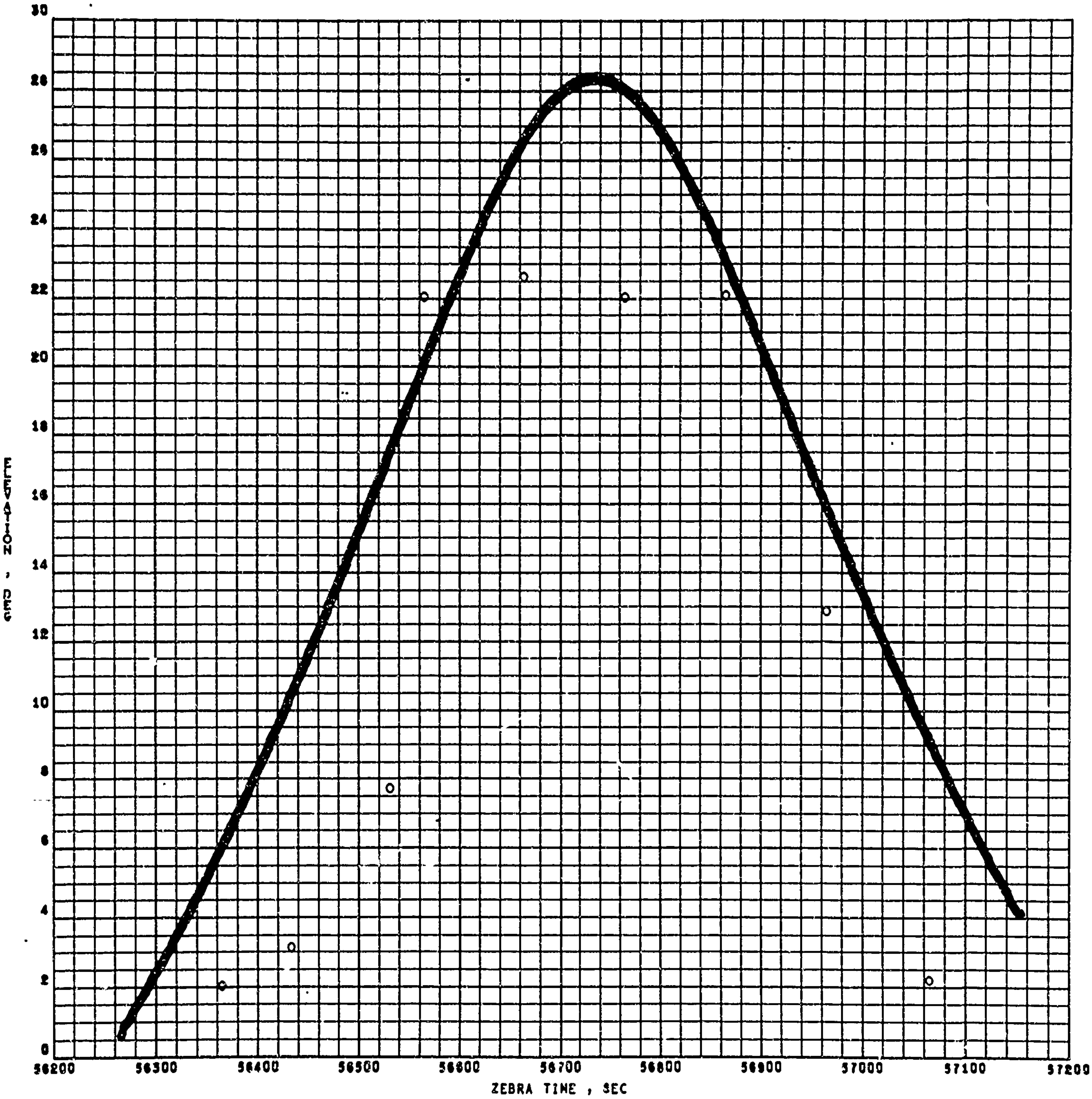


Figure 4 FIRST RAW DATA FILE
ELEVATION VERSUS TIME
(Echo II, pass 6069)

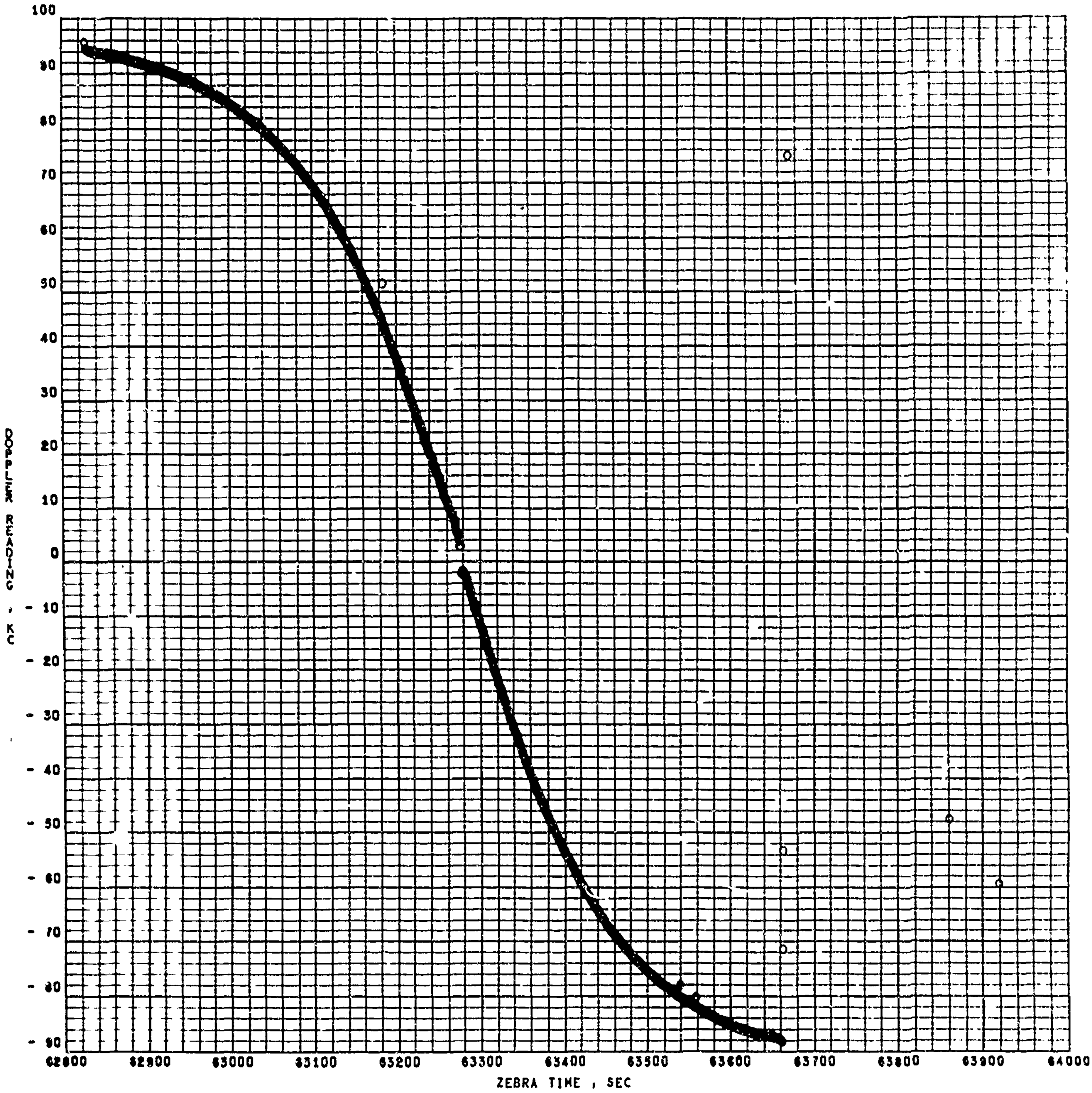


Figure 5 SECOND RAW DATA FILE
DOPPLER READING VERSUS TIME
(Echo II, pass 6070)

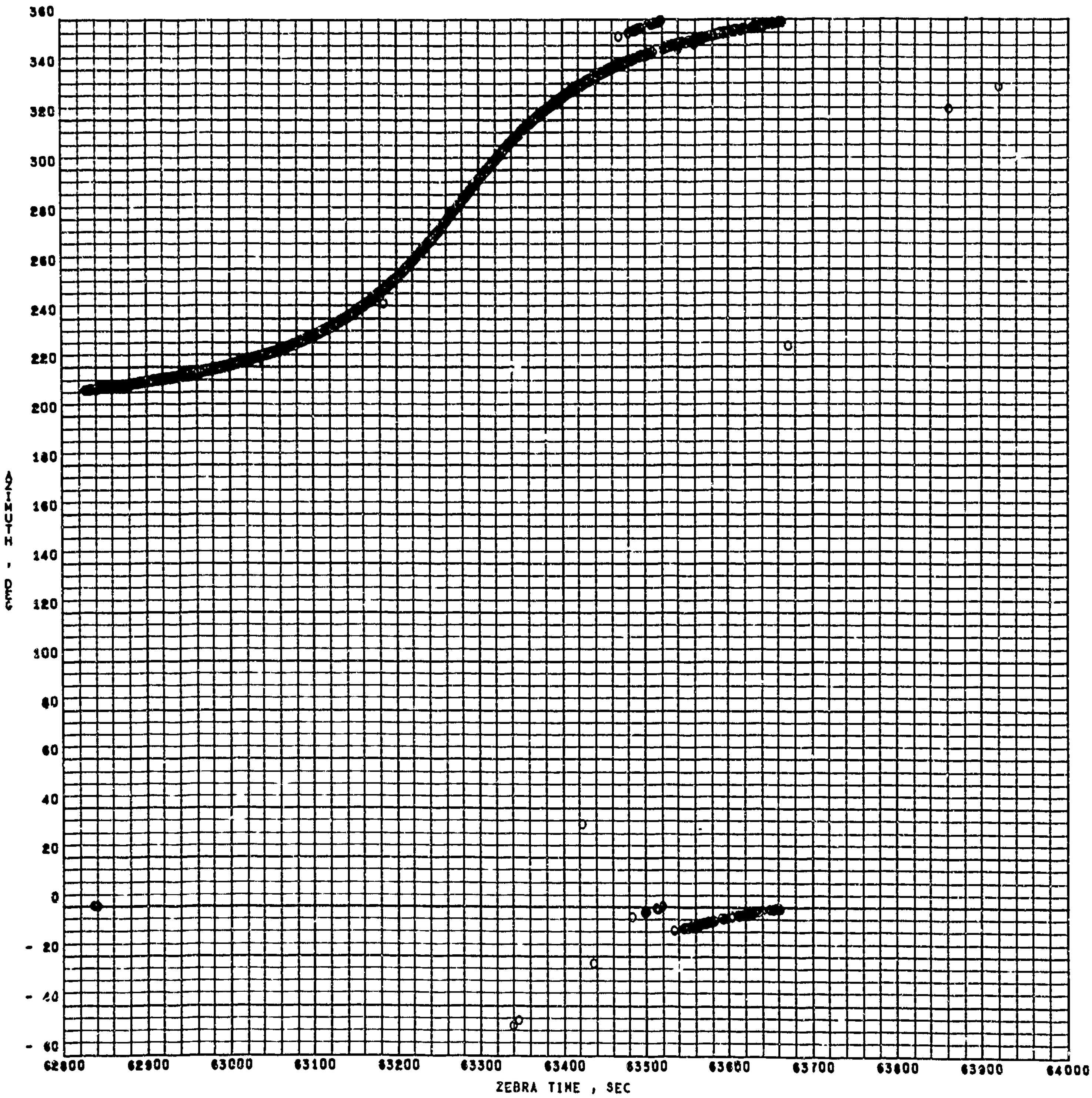


Figure 6 SECOND RAW DATA FILE
AZIMUTH VERSUS TIME
(Echo II, pass 6070)

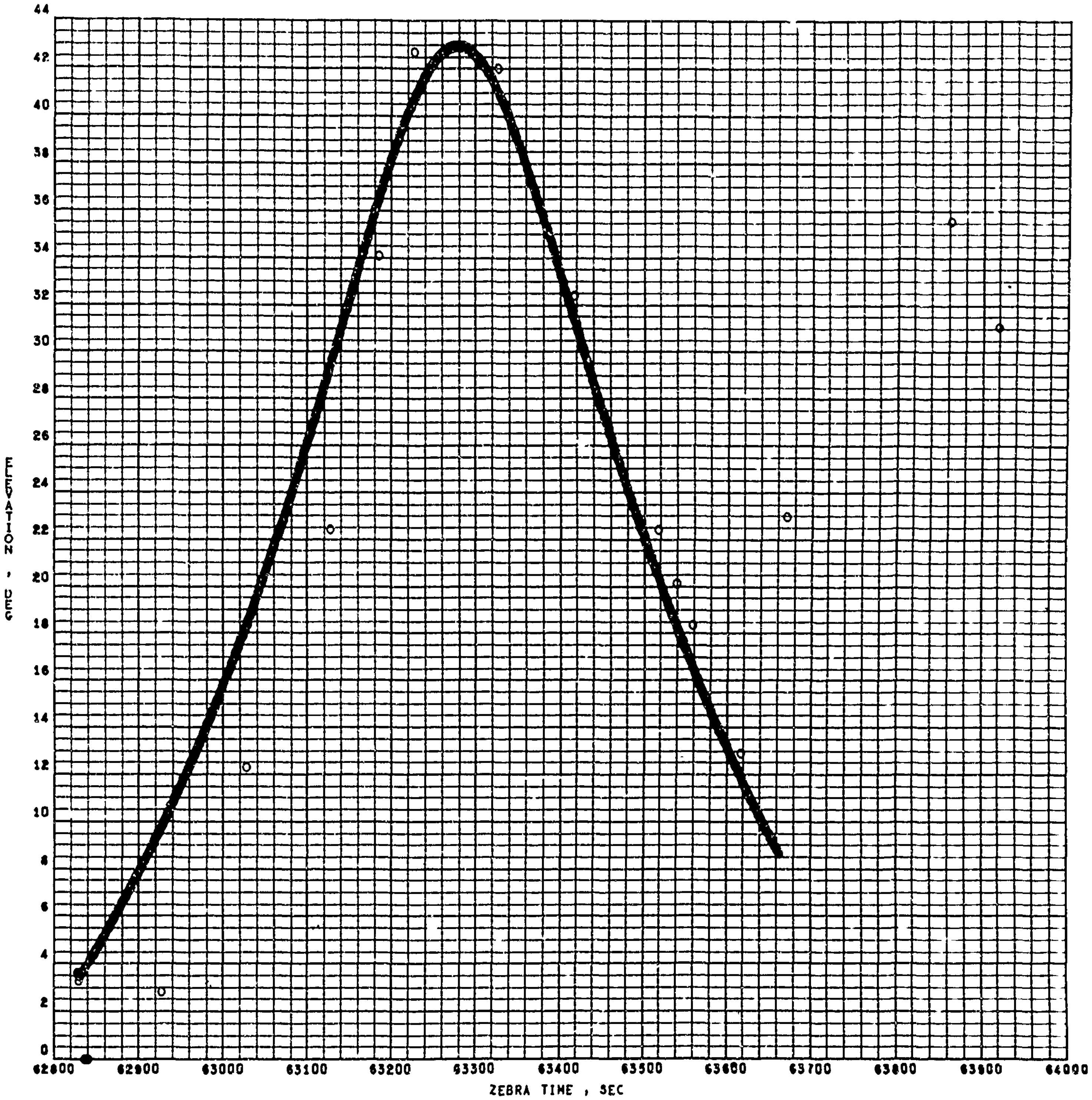


Figure 7 SECOND RAW DATA FILE
ELEVATION VERSUS TIME
(Echo II, pass 6070)

4.3.2 Variable Dimensions for Preprocessor

The required values for the variable dimensions were determined by the procedure outlined below. (See "Variable Dimensions" and "Storage Limitations" in the "Program Operation" section of SID 65-1203-2)

AA(4, MAXAA): This array must be large enough to hold the smoothed points corresponding to a single raw data file. The largest raw data file contains 3564 words. The number of words per information record is 4, and there are 4 words per data point (time, doppler reading, azimuth, elevation.) Thus, the number of raw data points per file is (3564-4)/4 = 890. Now the smoothing routine reduces 20 raw data points to a single smoothed point, therefore, since 890/20 45, MAXAA was dimensioned by 50.

A(6, MAXA): The primary purpose of this array is to receive the smoothed data from the AA array. Since MAXAA was set to 50 and there are two files of data, MAXA was set to 2 x 50 = 100. If there had been storage limitations, MAXA could have been reduced (the smoothed points would then be temporarily stored on tape). However, this array must be large enough to hold the smoothed points corresponding to a single raw data file.

STN (6, MAXSTN): The usual criterion for dimensioning this array is number of raw data points per file. The sample problem has MAXSTN = 1000.

This procedure is outlined here since more demanding tasks will undoubtedly be presented to the program during its period of use. The sample, in no way taxes the capability of the program. Before passing it is noted that these variables can be assigned large values to assure operation for most problems and that should these limits ever be violated a message will be printed as to the nature of the problem and steps required to correct it.

4.3.3 Input Data Load Sheet for the Preliminary Processor

FORTRAN FIXED 10 DIGIT DECIMAL DATA

DECK NO. DATA PROGRAMMER J. DOE DATE ______ PAGE 1 of 1 JOB NO. ______

Col.	NUMBER	IDENTIFICATION	DESCRIPTION DO NOT KEY PUNCH
1	0.1		PREPROCESSOR OUTPUT OPTIONS
13			SET TO PROVIDE ONLY THE SMOOTHED
25			DATA
37			
49		73 80	
61		DATA 1	
1	1		STATION I.D.
13	1.0		STATION SORTING ORDER
25	1.0		RANGE CONVERSION (not used)
37	-0.066033359		RANGE-RATE CONVERSION TO KM/SEC.
49	0.0	73 80	
61	0.017453292	DATA 2	AZIMUTH AND ELEVATION DEG TO RAG
1			END OF DATA
13			
25			
37			
49		73 80	
61		DATA 3	
1			
13			
25			
37			
49		73 80	
61			

Form 114-0-17 Rev. 7-58 (Vellum)

4.3.4 Preprocessor Output

Primary output of the preprocessor is a magnetic tape read directly by the differential corrections program (a format shown in Figure 8). However, several print options exist (see SID 65-1203-2). These options include

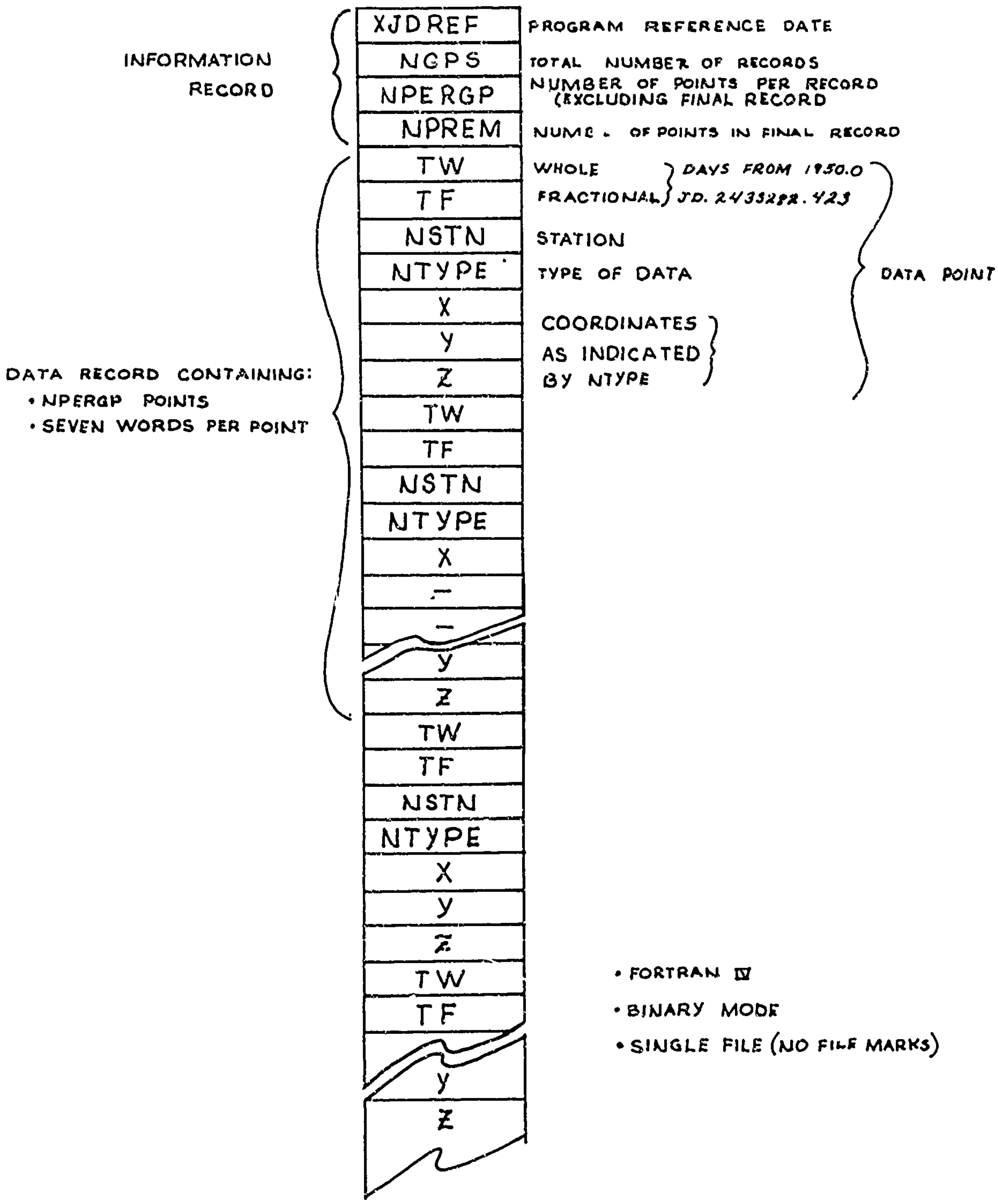
printing of the raw data
printing of smoothed data

Since the raw data were provided in graphical form on previous pages, the first option was not mechanized. The second option was however employed.

Accordingly, smoothed and sorted data corresponding to the curves presented earlier are presented in the following table.

FIGURE 8

FS4-305 A
OUTPUT TAPE FORMAT
SMOOTHED AND ORDERED DATA



*** SMOOTHED AND SORTED DATA ***   REFERENCE DATE IS 2433282.423

LINE NO	TIME (DAYS FROM REF DATE) WHOLE	FRAC	STN	TYPE	X	Y	Z
1	0.55950000E 04	0.72833095E 00	1	6	-0.59763414E 01	0.26872827E 01	0.20425821E-01
2	0.55950000E 04	0.72856221E 00	1	6	-0.59238771E 01	0.26688221E 01	0.37454991E-01
3	0.55950000E 04	0.72879396E 00	1	6	-0.58662044E 01	0.26493267E 01	0.55964530E-01
4	0.55950000E 04	0.72902522E 00	1	6	-0.57974219E 01	0.26290999E 01	0.75002535E-01
5	0.55950000E 04	0.72925649E 00	1	6	-0.57199250E 01	0.26066798E 01	0.99500826E-01
6	0.55950000E 04	0.72948823E 00	1	6	-0.56289957E 01	0.25825329E 01	0.11606177E 00
7	0.55950000E 04	0.72973142E 00	1	6	-0.55197529E 01	0.25546699E 01	0.13919396E 00
8	0.55950000E 04	0.72996268E 00	1	6	-0.54055400E 01	0.25264644E 01	0.16143649E 00
9	0.55950000E 04	0.73019443E 00	1	6	-0.52724449E 01	0.24952876E 01	0.18459263E 00
10	0.55950000E 04	0.73042569E 00	1	6	-0.51208623E 01	0.24613850E 01	0.20883031E 00
11	0.55950000E 04	0.73065696E 00	1	6	-0.49402078E 01	0.24243607E 01	0.23355949E 00
12	0.55950000E 04	0.73088870E 00	1	6	-0.47447134E 01	0.23834637E 01	0.25918019E 00
13	0.55950000E 04	0.73111997E 00	1	6	-0.45224100E 01	0.23388626E 01	0.28519265E 00
14	0.55950000E 04	0.73135171E 00	1	6	-0.42674593E 01	0.22894212E 01	0.31143584E 00
15	0.55950000E 04	0.73158298E 00	1	6	-0.39856571E 01	0.22352178E 01	0.33689001E 00
16	0.55950000E 04	0.73181472E 00	1	6	-0.36670021E 01	0.21759135E 01	0.36429025E 00
17	0.55950000E 04	0.73204599E 00	1	6	-0.33107943E 01	0.21106403E 01	0.38977195E 00
18	0.55950000E 04	0.73227773E 00	1	6	-0.29186757E 01	0.20397824E 01	0.41424826E 00
19	0.55950000E 04	0.73250899E 00	1	6	-0.24989965E 01	0.19625130E 01	0.43683925E 00
20	0.55950000E 04	0.73274074E 00	1	6	-0.20506103E 01	0.18795771E 01	0.45874582E 00
21	0.55950000E 04	0.73297200E 00	1	6	-0.15723250E 01	0.17915597E 01	0.47298806E 00
22	0.55950000E 04	0.73320327E 00	1	6	-0.10990717E 01	0.16992925E 01	0.48529038E 00
23	0.55950000E 04	0.73343501E 00	1	6	-0.64958085E 00	0.16037576E 01	0.49264039E 00
24	0.55950000E 04	0.73366628E 00	1	6	0.15468145E 00	0.15069345E 01	0.49501660E 00
25	0.55950000E 04	0.73389802E 00	1	6	0.63444376E 00	0.14105670E 01	0.49512568E 00
26	0.55950000E 04	0.73412929E 00	1	6	0.11644897E 01	0.13168006E 01	0.48406604E 00
27	0.55950000E 04	0.73436103E 00	1	6	0.16726395E 01	0.12258930E 01	0.47147911E 00
28	0.55950000E 04	0.73459230E 00	1	6	0.21524288E 01	0.11402630E 01	0.45496891E 00
29	0.55950000E 04	0.73482404E 00	1	6	0.25980171E 01	0.10602043E 01	0.43529823E 00
30	0.55950000E 04	0.73505531E 00	1	6	0.30042937E 01	0.98582216E 00	0.41395060E 00
31	0.55950000E 04	0.73528657E 00	1	6	0.33753208E 01	0.91750271E 00	0.38947085E 00
32	0.55950000E 04	0.73551831E 00	1	6	0.37063770E 01	0.85532304E 00	0.36470627E 00
33	0.55950000E 04	0.73574958E 00	1	6	0.40022906E 01	0.79839659E 00	0.33960551E 00
34	0.55950000E 04	0.73598132E 00	1	6	0.42631195E 01	0.74709015E 00	0.31380089E 00
35	0.55950000E 04	0.73621259E 00	1	6	0.44898364E 01	0.70046204E 00	0.29026785E 00
36	0.55950000E 04	0.73644433E 00	1	6	0.46992580E 01	0.65765197E 00	0.26361281E 00
37	0.55950000E 04	0.73667560E 00	1	6	0.48800085E 01	0.61888137E 00	0.23909131E 00

*** SMOOTHED AND SORTED DATA ***   REFERENCE DATE IS 2433282.423

LINE NO	TIME (DAYS FROM REF DATE) WHOLE	FRAC	STN	TYPE	X	Y	Z
38	0.55950000E 04	0.73690734E 00	1	6	0.50347922E 01	0.58365851E 00	0.21534868E 00
39	0.55950000E 04	0.73713861E 00	1	6	0.51708542E 01	0.55149679E 00	0.19217172E 00
40	0.55950000E 04	0.73737035E 00	1	6	0.52970178E 01	0.52235375E 00	0.17329870E 00
41	0.55950000E 04	0.73760162E 00	1	6	0.53998458E 01	0.49548568E 00	0.14792131E 00
42	0.55950000E 04	0.73783288E 00	1	6	0.54805636E 01	0.47070983E 00	0.12684373E 00
43	0.55950000E 04	0.73806462E 00	1	6	0.55731775E 01	0.44799975E 00	0.10639758E 00
44	0.55950000E 04	0.73829589E 00	1	6	0.56384577E 01	0.42698199E 00	0.87028387E-01
45	0.55950000E 04	0.73845802E 00	1	6	0.47999006E 01	0.41334471E 00	0.73844969E-01
46	0.55950000E 04	0.80430303E 00	1	6	-0.62040303E 01	0.36879234E 01	0.68141043E-01
47	0.55950000E 04	0.80455766E 00	1	6	-0.61606917E 01	0.37050498E 01	0.91884009E-01
48	0.55950000E 04	0.80478892E 00	1	6	-0.61131650E 01	0.37214869E 01	0.11456150E 00
49	0.55950000E 04	0.80502067E 00	1	6	-0.60542322E 01	0.37391791E 01	0.13802219E 00
50	0.55950000E 04	0.80525193E 00	1	6	-0.59883771E 01	0.37585956E 01	0.15660166E 00
51	0.55950000E 04	0.80548368E 00	1	6	-0.59120244E 01	0.37800339E 01	0.18916610E 00
52	0.55950000E 04	0.80571494E 00	1	6	-0.58203812E 01	0.38032021E 01	0.21659139E 00
53	0.55950000E 04	0.80594669E 00	1	6	-0.57125384E 01	0.38292371E 01	0.24571757E 00
54	0.55950000E 04	0.80617795E 00	1	6	-0.55882421E 01	0.38576692E 01	0.27647245E 00
55	0.55950000E 04	0.80640969E 00	1	6	-0.54421175E 01	0.38899242E 01	0.30356655E 00
56	0.55950000E 04	0.80664096E 00	1	6	-0.52756816E 01	0.39260563E 01	0.34344384E 00
57	0.55950000E 04	0.80687223E 00	1	6	-0.50784546E 01	0.39667442E 01	0.37973947E 00
58	0.55950000E 04	0.80710397E 00	1	6	-0.48482768E 01	0.40134528E 01	0.41825418E 00
59	0.55950000E 04	0.80733524E 00	1	6	-0.45803471E 01	0.40661716E 01	0.45826305E 00
60	0.55950000E 04	0.80756698E 00	1	6	-0.42682014E 01	0.41269672E 01	0.49376079E 00
61	0.55950000E 04	0.80779824E 00	1	6	-0.39054744E 01	0.41969820E 01	0.54299332E 00
62	0.55950000E 04	0.80802999E 00	1	6	-0.34619236E 01	0.42834608E 01	0.58864354E 00
63	0.55950000E 04	0.80826125E 00	1	6	-0.30108971E 01	0.43715266E 01	0.62847697E 00
64	0.55950000E 04	0.80849300E 00	1	6	-0.24824771E 01	0.44785511E 01	0.66781978E 00
65	0.55950000E 04	0.80872426E 00	1	6	-0.19055064E 01	0.45997658E 01	0.70386114E 00
66	0.55950000E 04	0.80895600E 00	1	6	-0.13050883E 01	0.47339210E 01	0.72862721E 00
67	0.55950000E 04	0.80918727E 00	1	6	-0.72142515E 00	0.48780511E 01	0.74500310E 00
68	0.55950000E 04	0.80941854E 00	1	6	0.12533200E 00	0.50271512E 01	0.74983505E 00
69	0.55950000E 04	0.80965028E 00	1	6	0.79846400E 00	0.51749880E 01	0.74226862E 00
70	0.55950000E 04	0.80988155E 00	1	6	0.14500388E 01	0.53157347E 01	0.72451431E 00
71	0.55950000E 04	0.81011329E 00	1	6	0.20561499E 01	0.54450393E 01	0.69566452E 00
72	0.55950000E 04	0.81034455E 00	1	6	0.26182286E 01	0.55605327E 01	0.66102117E 00
73	0.55950000E 04	0.81057630E 00	1	6	0.31154039E 01	0.56620597E 01	0.62189667E 00
74	0.55950000E 04	0.81080756E 00	1	6	0.35513442E 01	0.57508807E 01	0.58119433E 00

*** SMOOTHED AND SORTED DATA *** REFERENCE DATE IS 2433282.423

LINE NO	TIME (DAYS FROM REF DATE) WHOLE	FRAC	STN	TYPE	X	Y	Z
75	0.55950CCCE 04	0.81103931E 00	1	6	0.39376059E 01	0.58274211E 01	0.54065302E 00
76	0.55950000E 04	0.81116662E 00	1	6	0.41232708E 01	0.58652009E 01	0.51683888E 00
77	0.5595000CE 04	0.81139789E 00	1	6	0.44255949E 01	0.59266864E 01	0.47645787E 00
78	0.55950000E 04	0.81162915E 00	1	6	0.46824166E 01	0.59959253E 01	0.43748304E 00
79	0.5595CC00E 04	0.81186090E 00	1	6	0.48986638E 01	0.61105295E 01	0.40029511E 00
80	0.5595000CE 04	0.81209216E 00	1	6	0.50924568E 01	0.62233641E 01	0.36594906E 00
81	0.55950C00E 04	0.81232391E 00	1	6	0.52459573E 01	0.61067523E 01	0.33008936E 00
82	0.5595000CE 04	0.81255517E 00	1	6	0.53777257E 01	0.61381948E 01	0.29965509E 00
83	0.55950C0CE 04	0.81278691E 00	1	6	0.54960118E 01	0.61675483E 01	0.26967902E 00
84	0.55950C00E 04	0.81301818E 00	1	6	0.55948447E 01	0.61941655E 01	0.24111215E 00
85	0.559500C0E 04	0.81324992E 00	1	6	0.56793135E 01	0.62182120E 01	0.21525281E 00
86	0.559500C0CE 04	0.81348119E 00	1	6	0.57369688E 01	0.62402965E 01	0.18843286E 00
87	0.55950C0CE 04	0.81371293E 00	1	6	0.57772977E 01	0.62601354E 01	0.16388562E 00
88	0.55950000E 04	0.81384025E 00	1	6	0.47101759E 01	0.62716799E 01	0.15126244E 00
89	0.55950CCCE 04	0.81385169E 00	1	6	0.35057233E 01	0.62730221E 01	0.15083309E 00

DUMP NUMBER 000C1

4.4 Sample Differential Correction Program Input

The input for the sample problem will be discussed in relation to the significance of the data and its location in the fundamental data array. All quantities which can be readily changed will be discussed though some, as will be noted, will not be changed from their preprogrammed values.

Reference to the map of the fundamental array (Appendix 1) will varify the DATA locations given in the following pages.

4.4.1 Physical Constants for Math Model

Equatorial Radius	DATA (1)	As Programmed
Polar Radius	DATA (2)	As Programmed
Coefficient J	DATA (3)	As Programmed
H	DATA (4)	As Programmed
D	DATA (5)	As Programmed
GM for the Earth	DATA (6)	As Programmed
Spin Rate for the Earth	DATA (7)	As Programmed
GM for the Moon	DATA (8)	As Programmed
GM for the Sun	DATA (9)	As Programmed
The astronomical Unit	DATA (10)	As Programmed

4.4.2 Satellite Data for ECHO II

Mass (assumed)	DATA (16)	1.0
Area (assumed)	DATA (17)	100.0
Drag Coefficient (")	DATA (18)	2.0
Reflectivity (")	DATA (19)	1.0
Position Vector X	DATA (20)	4952.3943
Y	DATA (21)	1406.9609
Z	DATA (22)	-5362.9226
Velocity Vector X	DATA (23)	4.4573218
Y	DATA (24)	2.9062537
Z	DATA (25)	5.0928345

	Time Relative to 1950.0
0^h April 27 1965 =	2438877.5
Epoch =	.63865740
J.D. of Epoch	2438878.13865740
Reference Epoch	2433282.423
	5595.71565740 days

TW	DATA (26)	5595. days
TF	DATA (27)	.71565740 days

4.4.3 Control Options

WINDEX	DATA (28)	1.
CHECK	DATA (29)	1.
GONO	DATA (30)	1.
CODUMP	DATA (31)	1.

4.4.4 Station Identification Card (SIC)

Station one (Floyd) will be utilized
Stations two through ten will be deleted

4.4.5 Station Data*

Latitude	(rad)	DATA (36)	.75393225
Longitude	(rad)	DATA (37)	4.96824718
Altitude	(Km)	DATA (38)	.17937
Station Name		DATA (39)	As Programmed
Horizon Correction		DATA (77)	As Programmed
Variance in Latitude		DATA (87)	As Programmed
Longitude		DATA (88)	As Programmed
Altitude		DATA (89)	As Programmed
Range		DATA (176)	As Programmed
Range-rate		DATA (177)	.01 **
Azimuth		DATA (178)	As Programmed
Elevation		DATA (179)	As Programmed

4.4.6 COVARIANCE MATRIX FOR ERRORS IN $\vec{r}$ AND $\vec{v}$

The data for the elements imply that estimates of the errors in these parameters are approximately:

$$3\sigma_{\Delta a} = .1 \text{ Km}$$
$$3\sigma_{\Delta e} = .0001$$
$$3\sigma_{\Delta i} = \frac{0.01}{57.3} \text{ rad}$$
$$3\sigma_{\Delta M} = \frac{0.01}{57.3} \text{ rad}$$
$$3\sigma_{\Delta \omega} = \frac{0.01}{57.3} \text{ rad}$$
$$3\sigma_{\Delta \Omega} = \frac{0.01}{57.3} \text{ rad}$$

or

$$\sigma^2_{\Delta a} = 1 \times 10^{-3}$$
$$\sigma^2_{\Delta e} = 1 \times 10^{-9}$$
$$\sigma^2_{\Delta i} = .35 \times 10^{-8}$$
$$\sigma^2_{\Delta M} = .35 \times 10^{-8}$$
$$\sigma^2_{\Delta \omega} = .35 \times 10^{-8}$$
$$\sigma^2_{\Delta \Omega} = .35 \times 10^{-8}$$

* See subroutine INPUT and Appendix 1.

** From conversations with RADC personnel it was learned that the Doppler data from which $\dot{R}$ was computed were not as precise as had been assumed earlier. No firm estimate of variance was available, however.

Now, since this set of errors is relatable directly to the set of errors $d\vec{r}$ and $d\vec{v}$ in rotating coordinates by the matrix equation of Table 1, and since the rotating and inertial coordinates are relatable by a linear transformation, the covariance matrix for $d\vec{r}$ and $d\vec{v}$ can be evaluated as follows:

$$\{\Delta X\} = A\{\Delta E\}$$

$$\{\Delta X'\} = T\{\Delta X\} = TA\{\Delta E\}$$

$$\mathcal{E}(\Delta X' \Delta X'^T) = TA\,\mathcal{E}(\Delta E \Delta E^T)\,A^T T^T$$

where $\Delta X = \begin{Bmatrix} d\vec{r} \\ d\vec{v} \end{Bmatrix}$ rotating coordinates

$\Delta X' = \begin{Bmatrix} d\vec{r} \\ d\vec{V} \end{Bmatrix}$ cartesian coordinates

A = matrix of Table 1

$\Delta E = \begin{Bmatrix} \Delta a \\ \Delta e \\ \vdots \end{Bmatrix}$

T = Transformation from rotating to inertial coordinates

$$\Delta M = -\sqrt{\frac{\mu}{a^3}}\,\Delta t_p$$

But $\mathcal{E}(\Delta X' \Delta X'^T)$ is not the matrix required because it is valid for the initial epoch rather than the updated epoch. Thus, one final transformation is required

$$\Delta X_2' = \varphi(t_2, t_1)\,\Delta X_1'$$

and

$$\mathcal{E}(\Delta X_2' \Delta_2'^T) = \varphi(t_2, t_1)\,\mathcal{E}(\Delta X_1' \Delta X_1'^T)\,\varphi^T(t_2, t_1)$$

where $\varphi(t_2, t_1)$ is the state transition matrix. For the present purposes $\varphi(t_2, t_1)$ can be assumed to be adequately approximated using a conic reference trajectory (see subroutine TRANS).

Table 1

POSITION - VELOCITY SENSITIVITIES FOR ELLIPTIC ORBITS

$$\begin{bmatrix} dr \\ r\,d\varphi \\ r\,d\psi \\ dv \\ V\,d\gamma \\ V\,d\beta \end{bmatrix} = \begin{bmatrix} \frac{r}{a} - \frac{3}{2}\frac{aeM}{r}\sin E & -a\cos E + \frac{a^2 e}{r}\sin E & \frac{-a^2 e n}{r}\sin E & 0 & 0 & 0 \\ -\frac{3}{2}\frac{pM}{r(1-e^2)^{1/2}} & \frac{a\sin E}{(1-e^2)^{\frac{1}{2}}}(1+p/r) & -\frac{pan}{r(1-e^2)^{1/2}} & 0 & r & 0 \\ 0 & 0 & 0 & \frac{r\sin i}{\sin L}\left[\operatorname{ctn}^2 i \operatorname{ctn}^2\beta - 1\right] & 0 & r\cos L\cos\beta \\ -\frac{V}{2a} + \frac{3}{2}\frac{eaM\mu}{r^3 V}\sin E & \frac{\mu a\cos E}{r^2 V} - \frac{\mu a^2 e}{r^3 V}\sin^2 E & \frac{\mu a^2 e n}{r^3 V}\sin E & 0 & 0 & 0 \\ -\frac{3}{2}\frac{VaM(1-e^2)^{\frac{1}{2}}B}{r^2} & \frac{-aV\cos\theta B}{r\tan\gamma} + \frac{eV}{\tan\gamma(1-e^2)} & \frac{-Va^2 n(1-e^2)^{\frac{1}{2}}B}{r^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & -V\cos\beta\left[\frac{\sin i}{\cos L} + \operatorname{ctn} i\sin\beta\right] & 0 & -V\sin L\sin^2\beta \end{bmatrix} \begin{bmatrix} da \\ de \\ dt_p \\ di \\ d\omega \\ d\Omega \end{bmatrix}$$

$M = n(t - t_p)$ r = radius   E = eccentric anomaly   i = inclination

$B = 1 - \mu / rV^2$ v = velocity   a = semi-major axis θ = true anomaly

L = latitude γ = flight path angle   e = eccentricity

β = azimuth   p = semi-latus rectum

Adopting the notation $T_x(\alpha)$ etc. to mean a positive rotation about the X axis through the angle α, the transformation of position errors becomes

$$d\vec{r}\,' = T_z(-e)\, T_x(-i)\, T_z[-(\theta+\omega)]\, d\vec{r}$$

$$\equiv R(\Omega, i, \varphi)\, d\vec{r}$$

similarly

$$d\vec{v}\,' = T_z(-\Omega)\, T_x(-i)\, T_z[-(\theta+\omega)]\, T_z(-90+\gamma)\, d\vec{v}$$

$$\equiv R(\Omega, i, \Xi)\, d\vec{v}$$

where

$$\Xi = -90 - (\theta + \omega - \gamma)$$

thus, T becomes

$$T = \begin{bmatrix} R(e, i, \varphi) & 0 \\ 0 & R(e, i, \Xi) \end{bmatrix}$$

Therefore, in successive steps, the matrix A is

.96791741E 00	-.74851830E 04	-.19037164E-01	.00000000E-38	.00000000E-38	.00000000E-38
-.30336841E 01	.15902172E 04	-.74557372E 01	.00000000E-38	.73450973E 04	.00000000E-38
.00000000E-38	.00000000E-38	.00000000E-38	-.47498852E 04	.00000000E-38	.55403090E 04
-.48750563E-03	.74174433E 01	.18864880E-04	.00000000E-38	.00000000E-38	.00000000E-38
-.73161020E-04	.79840893E 00	-.17980426E-03	.00000000E-38	.00000000E-38	.00000000E-38
.00000000E-38	.00000000E-38	.00000000E-38	-.96218643E 01	.00000000E-38	-.17830626E 00

The matrix T is

-.77797139E 00	-.36374413E 00	.51229940E 00	.00000000E-38	.00000000E-38	.00000000E-38
-.38374998E 00	-.37052683E 00	-.84584028E 00	.00000000E-38	.00000000E-38	.0C000000E-38
.49749013E 00	-.85463445E 00	.14867248E 00	.00000000E-38	.00000000E-38	.00000000E-38
.00000000E-38	.00000000E-38	.00000000E-38	-.38099555E 00	.76966985E 00	.51229940E 00
.00000000E-38	.00000000E-38	.00000000E-38	-.37898896E 00	.37539520E 00	-.84584028E 00
.00000000E-38	.00000000E-38	.00000000E-38	-.84333251E 00	-.51641721E 00	.14867248E 00

The matrix $\varphi(t_2, t_1)$ is

.31543679E 02	.16143082E 02	-.19915249E 02	.16161649E 05	.16155360E 05	.34866917E 05
.26864938E 03	.13183176E 03	-.17063167E 03	.13796138E 06	.13640126E 06	.30287280E 06
.14166656E 04	.69946113E 03	-.90303723E 03	.72785634E 06	.72259516E 06	.16025917E 07
-.11454470E 01	-.56555799E 00	.73079533E 00	-.58835202E 03	-.58357696E 03	-.12960317E 04
-.63281701E 00	-.31196560E 00	.40339436E 00	-.32440901E 03	-.32316204E 03	-.71533289E 03
.34813951E 00	.17163150E 00	-.22275929E 00	.17844926E 03	.17771485E 03	.39522150E 03

The covariance matrix for the initial ($t = t_1$) errors in the inertial coordinate system is:

.10150787E 00	-.43109424E-01	.47253023E-01	.29521933E-04	-.80932066E-04	-.22894314E-04
-.43109424E-01	.16503310E 00	.26999317E-01	-.72766497E-04	.10635378E-03	-.34567335E-04
.47253023E-01	.26999316E-01	.17717868E 00	.23561947E-04	-.64302088E-05	.38728416E-04
.29521933E-04	-.72766498E-04	.23561947E-04	.89978761E-07	-.13488413E-06	.39491969E-07
-.80932066E-04	.10635378E-03	-.64302088E-05	-.13488413E-06	.23823809E-06	-.23945993E-07
-.22894314E-04	-.34567335E-04	.38728416E-04	.39491969E-07	-.23945993E-07	.51824178E-07

and the covariance matrix for the errors in the inertial coordinate system for $t = t_2$ is:

.45852944E 00	.31658305E 01	.17489094E 02	-.14051295E-01	-.78504921E-02	.43026154E-02
.31658304E 01	.32695791E 02	.17336036E 03	-.14017666E 00	-.77136100E-01	.43270175E-01
.17489095F 02	.17336038E 03	.92633961E 03	-.74853903E 00	-.41270284E 00	.23150944E 00
-.14051295E-01	-.14017667E 00	-.74853902E 00	.60501508E-03	.33342622E-03	-.18713211F-03
-.78504923E-02	-.77136103E-01	-.41270283E 00	.33342621E-03	.18400949E-03	-.10318048E-03
.43026156E-02	.43270178E-01	.23150944E 00	-.18713211E-03	-.10318049E-03	.58000710E-04

These data must be read into the differential corrections program column-wise. (The digital computer stores elements of A_{ij} in a single subscripted array $a_{11}, a_{21}, a_{31} \cdots a_{n1}; \; a_{12}, a_{22}, a_{32} \cdots a_{n2}$; etc.), thus

$$\begin{aligned} P_{11} &= \text{DATA } (355) \\ P_{21} &= \text{DATA } (356) \\ P_{31} &= \text{DATA } (357) \\ P_{41} &= \text{DATA } (358) \\ P_{51} &= \text{DATA } (359) \\ P_{61} &= \text{DATA } (360) \\ P_{12} &= \text{DATA } (361) \\ &\vdots \end{aligned}$$

4.4.7 Input Data Load Sheets for the Sample Problem

As was noted in the discussion of subroutine INPUT, most of the data required for program operation can be prerecorded (in fact data for a check network is presently provided within the program). As was also noted in the discussion of INPUT and REED, these data will not be affected by inputting the desired data unless new data are assigned to the locations presently occupied by these numbers in the common array (see Appendix 1). Thus, only the following data are required (the load sheets are in the proper format and should be copied).

FORTRAN FIXED 10 DIGIT DECIMAL DATA

DECK NO. ____ PROGRAMMER ____ DATE ____ PAGE 1 of 4 JOB NO. ____

	NUMBER	IDENTIFICATION	DESCRIPTION DO NOT KEY PUNCH
1	S I.C. 1 0		This card (The SIC) must
13	0 0 0		be the first card in the
25	0 0 0		data deck after the system
37	0 0		control card identifying the data
49		73 80	for the sample This card indicates
61		DATA 1	that only The FLOYD terminal is operative
1	16		locations of first piece of data on cards in the data array
13	1.0		assumed mass of satellite (Kg)
25	100.		assumed area of satellite (m^2)
37	2.0		assumed drag coefficient
49	1.0	73 80	assumed reflectivity
61		DATA 2	
1	20		location
13	4664.3825		X component of position (Km)
25	1223.8631		Y component of position (Km)
37	-5664.1829		Z component of position (Km)
49		73 80	
61		DATA 3	
1	23		location
13	4.7481037		$\dot{x}$ component of velocity (km/sec)
25	2.9870220		$\dot{y}$ component of velocity (Km/sec)
37	4.7656496		$\dot{z}$ component of velocity (Km/sec)
49		73 80	
61		DATA 4	

~ 1-u-0-17 Rev. 7-55 (Vellum)

FORTRAN FIXED 10 DIGIT DECIMAL DATA

DECK NO. ______ PROGRAMMER ______ DATE 2 PAGE 4 of ______ JOB NO. ______

Col.	Sign	NUMBER	IDENTIFICATION	DESCRIPTION DO NOT KEY PUNCH	
1		26		location	
13		5.595		TW (days from 2433282.423	
25		.71588888		TE fraction of Day	
37		1.		WINDEX (r,v = cartesian)	
49		1.	73 … 80	CHECK (check tracking stations at each point on Trajectory)	
61		1.	DATA 5	GONO (write r,v each time)	
1		355		location	
13		45852944+00		P_{11}	Km^2
25		31658304+01		P_{21}	Km^2
37		17489095+02		P_{31}	Km^2
49	-	14051295-01	73 … 80	P_{41}	Km^2/sec
61	-	78504923-02	DATA 6	P_{51}	Km^2/sec
1		360		location	
13		43026156-02		P_{61}	Km^2/sec
25		31668305+01		P_{12}	Km^2
37		32695791+02		P_{22}	Km^2
49		17336038+03	73 … 80	P_{32}	Km^2
61	-	14017667+00	DATA 7	P_{42}	Km^2/sec
1		365		location	
13	-	77136103 01		P_{52}	Km^2/sec
25		43270178-01		P_{62}	Km^2/sec
37		17489094+02		P_{13}	Km^2
49		17336036+03	73 … 80	P_{23}	Km^2
61		92633961+03	DATA 8	P_{33}	Km^2

Form 114-C-17 Rev. 7-58 (Vellum)

FORTRAN FIXED 10 DIGIT DECIMAL DATA

DECK NO.______ PROGRAMMER______ DATE______ PAGE 3 of 4 JOB NO.______

Col.		NUMBER	IDENTIFICATION	DESCRIPTION DO NOT KEY PUNCH	
1		370		location	
13	−	7485390 2 +00		P_{43}	Km²/sec
25	−	4127028 3 +00		P_{53}	Km²/sec
37		2315094 4 +00		P_{63}	Km²/sec
49	−	1405129 5 −01	73 … 80	P_{14}	Km²/sec
61	−	1401766 6 +00	DATA 9	P_{24}	Km²/sec
1		375		location	
13	−	7485390 3 +00		P_{34}	Km²/sec
25		6050150 8 −03		P_{44}	Km²/sec²
37		3334262 1 −03		P_{54}	Km²/sec²
49	−	1871321 1 −03	73 … 80	P_{64}	Km²/sec²
61	−	7850492 1 −02	DATA 10	P_{15}	Km²/sec
1		380		location	
13	−	7713610 0 −01		P_{25}	Km²/sec
25	−	4127028 4 +00		P_{35}	Km²/sec
37		3334262 2 −03		P_{45}	Km²/sec²
49		1840094 9 −03	73 … 80	P_{55}	Km²/sec²
61	−	1031804 9 −03	DATA 11	P_{65}	Km²/sec²
1		385		location	
13		4302615 4 −02		P_{16}	Km²/sec
25		4327017 5 −01		P_{26}	Km²/sec
37		2315094 4 +00		P_{36}	Km²/sec
49	−	1871321 1 −03	73 … 80	P_{46}	Km²/sec²
61	−	1031804 8 −03	DATA 12	P_{56}	Km²/sec²

Form 111-C-17 Rev. 7-58 (Vellum)

FORTRAN FIXED 10 DIGIT DECIMAL DATA

DECK NO. ______ PROGRAMMER ______ DATE ______ PAGE 4 of 4 JOB NO. ______

	NUMBER	IDENTIFICATION	DESCRIPTION DO NOT KEY PUNCH
1	39.0		location
13	5800071.0-04		P_{66} Km^2/sec^2
25			
37			
49		73 80	
61		DATA 13	
1	999		Control Card to indicate that numerical
13			data are complete and to initiate input
25			of station name data
37			
49		73 80	
61		DATA 14	
1	9999		Control Card to indicate that all data
13			have been provided
25			
37			
49		73 80	
61		DATA 15	
1			
13			
25			
37			
49		73 80	
61			

Form 114-C-17 Rev. 7-58 (Vellum)

Five notes are worthy of attention before this discussion ends.

1) The SIC must be the first physical card in the data deck.

2) The 999 control card must follow all numerical data.

3) The 9999 control card must follow the station name data even if there are no changes (as in the case of the sample).

4) The order of the cards following the SIC and preceding the 999 card can be arbitrary since a location is provided with each group of data.

5) With the exception of the locations and data provided on the SIC, all numbers are floating point. These floating point numbers can be expressed with a decimal point (as on card 2) or without (as on cards 6-13). However, in the latter case, a decimal is assumed to be located between the first two locations of the field (i.e. before the first digit) and the last 3 locations of the field must contain the exponent of 10 which properly defines the number in question.

4.4.8 Program Output

The differential corrections program was utilized in conjunction with the data of the preceding pages to compute the orbit of ECHO II. The results of this effort will be presented in their entirety to facilitate checkout on other systems and to demonstrate program operation.

However, before presenting these data, it is felt advisable to discuss several problems which were encountered. First, the trajectory, as computed from the data provided, checked the Goddard estimate of this time with negligible error and the observed values of azimuth and elevation to within one-half of a degree after seven days of prediction. This fact implied again a good level of accuracy. But, the computed values of range-rate failed to show this level of agreement. This fact was not considered entirely unreasonable at the time.

Second, reference to the raw data disclosed that the range-rate as recorded exhibited a discontinuity of approximately 2.6 Kc at the time the doppler changed signs. This jump corresponds to approximately 0.172 Km/sec, and thus could explain the range-rate disagreement which was experienced if it could be ascertained how this correction should be applied; (i.e., is the positive segment of doppler correct, the negative correct, or are both in error?). Regardless, however, the data are in error and will affect the results in two distinct ways:

1) Since the data are biased, the filter process, as coded, is not correct (biases are assumed to have been subtracted from the observed data).

2) Since the data are to be processed with the biases included, an oscillatory nature of the solution is to be expected (under the assumption that neither branch of the doppler curve is correct).

Third, since the range-rate data was assumed to be in error, a call was placed to RADC (in the person of Mr. Frank Braddley) to ascertain the level of accuracy previously obtained by RADC in range-rate measurements. No precise information was obtained, though it was indicated that the accuracy was subject to question. For this reason, reference was again made to the raw data to determine whether variance data earlier assumed for range-rate data were reasonable. The results of this review indicated that these earlier data should be replaced with an estimate of variance on the order of .0001 Km^2/sec^2. Therefore, these modifications to the data enumerated in the previous sections have been made.

At this point, refer to the sample problem presented on subsequent pages (pass 6069 of ECHO II) and note that

1) All data utilized in the program are printed for reference during checkout.

2) The operation of the trajectory, integration and tracking groups is demonstrated.

3) The behavior of the filter is demonstrated.

Now, note that after the trace of the covariance matrix for estimation errors has been reduced (the initial errors are quite large since they have propagated for approximately 8 days) and orbital elements begin to be printed for the rectified (corrected) orbit (U.T. = 56616. sec), there is a steady change in the elements "a" and "e." The first printed values lie somewhat below those predicted by Goddard (about 1% in semi-major axis and 10% in eccentricity), but as subsequent points are processed the solution is attained, passed, attained again, and finally passed for the second time.

As is apparent, the process is tending to converge, though the rate of convergence is less than originally anticipated. The observed rate of convergence is believed to be affected by two distinct factors. First, the nature of the doppler data itself will preclude exact convergence and may introduce an oscillatory motion in the solution. Second, the "memory" of the system is slight. In the way of explanation, the history (effect) of all previous data points is carried in the covariance matrix for the estimation errors. This matrix in turn was allowed to "deteriorate" for almost eight days following the last "fix" on the satellite in the sample problem though, in reality, more recent estimates of this matrix probably existed. Thus, while the trajectory itself could be updated with reasonable accuracy from the prescribed epoch of 20 April 1965, it is unlikely that a "true" data reduction problem would be subjected to errors as large as those utilized in the sample except for the initial phases of the problem (i.e., immediately following injection). Rather, it is expected that the estimation errors at the epoch of 27 April 1965 were probably similar to those constructed for the epoch of 20 April 1965. This change in the sample problem would reduce terms in the assumed covariance matrix by terms ranging from 10^3 to 10^4, and would reduce the effect of the differences in the observed and computed values of the data by the same factors, thus allowing the trajectory of ECHO II to be corrected with precision. To illustrate this conclusion, the reduced data for an epoch near the final data point is presented in the following table for this variation of the sample problem.

TABLE 2

ELLIPTIC ORBIT ELEMENTS

SEMIMAJOR AXIS	(KM)	=	0.75307251E 04
ECCENTRICITY		=	0.24977477E-01
TRUE ANOMOLY DATE	(DEG)	=	0.14263121E 01
R ASC OF ASC NODE	(DEG)	=	0.24846007E 02
ARG OF PERIAPSE	(DEG)	=	0.14930505E 02
ORBIT INCLINATION	(DEG)	=	0.81473733E 02
DEL NODE PER REV	(RAD)	=	-0.10861821E-02
DEL APSE PER REV	(RAD)	=	-0.32604282E-02

As is also apparent in the sample problem, the limit of the convergence process is not identical to the Goddard data. This fact is not alarming under the circumstances, due to the gross nature of the assumptions regarding the statistics of the problem, the poor quality of some of the data, and the small number of points processed. (The entire pass was reduced to approximately 45 points.) It is fully expected that a more complete sample problem involving more precisely reduced raw data will converge to the desired solution.

INPUT DATA

DATE(1950) =	0.55950000E 04	0.71565740E 00	
RADIUS VEC. =	0.49523943E 04	0.14069609E 04	-0.53629226E 04
VELOCITY VEC=	0.44573218E 01	0.29062537E 01	0.50928345E 01

INITIAL COVARIANCE MATRIX

0.45952444E 00	0.31658305E 01	0.17489094E 02	-0.14051295E-01	-0.78504921E-02	0.43026154E-02
0.31658304E 01	0.32695791E 02	0.17336036E 03	-0.14017666E 00	-0.77136100E-01	0.43270175E-01
0.17489095E 02	0.17336038E 03	0.92633961E 03	-0.74853903E 00	-0.41270284E 00	0.23150944E 00
-0.14051295E-01	-0.14017667E 00	-0.74853902E 00	0.60501508E-03	0.33342622E-03	-0.18713211E-03
-0.78504923E-02	-0.77136103E-01	-0.41270283E 00	0.33342621E-03	0.18400949E-03	-0.10318048E-03
0.43026156E-02	0.43270178E-01	0.23150944E 00	-0.18713211E-03	-0.10318049E-03	0.58000710E-04

SATELLITE DATA

MASS =	0.10000000E 01
AREA =	0.10000000E 03
CD =	0.20000000E 01
REFLECTIVITY=	0.10000000E 01

THE FOLLOWING STATIONS WILL BE CONSIDERED

FLOYD

*** DATA ARRAY ***

DATA	CON						
1	1	0.63781630E 04	0.63567770E 04	0.16234500E-02	-0.20500000E-05	0.92750000E-05	0.39860150E 06
7	7	0.72921165E-04	0.49007588E 04	0.13271545E 12	0.14953000E 09	0.86400000E 05	0.17453292E-01
13	13	0.00000001E-38	0.00000001E-38	0.00000000E-38			
	SAT						
16	1	0.10000000E 01	0.10000000E 03	0.20000000E 01	0.10000000E 01	0.49523943E 04	0.14069609E 04
22	7	-0.53629226E 04	0.44573218E 01	0.29062537E 01	0.50928345E 01	0.55950000E 04	0.71565740E 00
28	13	0.10000000E 01	0.10000000E 01	0.10000000E 01	0.10000000E 01	0.00000000E-38	0.00000000E-38
34	19	0.00000000E-38	0.00000000E-38				
	SDA						
36	1	0.75393225E 00	0.49682472E 01	0.17937480E 00	0.20308425E 16	0.00000000E-38	0.00000000E-38
42	7	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
48	13	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
54	19	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
60	25	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
66	31	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
72	37	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
78	43	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
84	49	0.00000000E-38	0.00000000E-38	0.10000000E-11	0.10000000E-11	0.10000000E-07	0.00000000E-38
90	55	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
96	61	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
102	67	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
108	73	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
114	79	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
120	85	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
126	91	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
132	97	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
138	103	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
144	109	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
150	115	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
156	121	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
162	127	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
168	133	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
174	139	0.00000000E-38	0.00000000E-38	0.22500000E-03	0.10000000E-03	0.40000000E-05	0.40000000E-05
180	145	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
186	151	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
192	157	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
198	163	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
204	169	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38

*** DATA ARRAY ***

DATA	SDA						
210	175	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
216	181	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
222	187	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
228	193	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
234	199	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
240	205	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
246	211	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
252	217	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
258	223	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
264	229	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
270	235	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
276	241	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
282	247	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38		
	STT						
286	1	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
292	7	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
298	13	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
304	19	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
310	25	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
316	31	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
322	37	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
328	43	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
334	49	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
340	55	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
346	61	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
352	67	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.45852944E 00	0.31658304E 01	0.17489095E 02
358	73	-0.14051295E-01	-0.78504923E-02	0.43026156E-02	0.31658305E 01	0.32695791E 02	0.17336038E 03
364	79	-0.14017667E 00	-0.77136103E-01	0.43270178E-01	0.17489094E 02	0.17336036E 03	0.92633961E 03
370	85	∓0.74853902E 00	-0.41270283E 00	0.23150944E 00	-0.14051295E-01	-0.14017666E 00	-0.74853903E 00
376	91	0.60501508E-03	0.33342621E-03	-0.18713211E-03	-0.78504921E-02	-0.77136100E-01	-0.41270284E 00
382	97	0.33342622E-03	0.18400949E-03	-0.10318049E-03	0.43026154E-02	0.43270175E-01	0.23150944E 00
388	103	-0.18713211E-03	-0.10318048E-03	0.58000710E-04			

*** DATA ARRAY ***

DATA	WRK						
391	1	0.99999333E 00	0.33455521E-02	0.14545751E-02	-0.33455706E-02	0.99999440E 00	0.99870099E-05
397	7	-0.14545326E-02	-0.14850885E-04	0.99999893E 00	0.99999333E 00	-0.33455706E-02	-0.14545326E-02
403	13	0.33455521E-02	0.99999440E 00	-0.14850885E-04	0.14545751E-02	0.99870099E-05	0.99999893E 00
409	19	0.10000000E 01	-0.78634635E-04	-0.34099568E-04	0.78634635E-04	0.10000000E 01	0.12419138E-04
415	25	0.34099568E-04	-0.12419138E-04	0.10000000E 01	0.49492675E 04	0.13903309E 04	-0.53701412E 04
421	31	0.44744230E 01	0.28913760E 01	0.50863025E 01	0.55950000E 04	0.71565740E 00	0.49523943E 04
427	37	0.14069609E 04	-0.53629226E 04	0.44573218E 01	0.29062537E 01	0.50928345E 01	0.55950000E 04
433	43	0.71565740E 00	0.49492675E 04	0.13903309E 04	-0.53701412E 04	0.44744230E 01	0.28913760E 01
439	49	0.50863025E 01	0.55950000E 04	0.71565740E 00	0.00000000E-38	0.00000000E-38	0.00000000E-38
445	55	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
451	61	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
457	67	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
463	73	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
469	79	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
475	85	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
481	91	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
487	97	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
493	103	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
499	109	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
505	115	0.00000000E-38	0.00000000E-38	0.49492675E 04	0.13903309E 04	-0.53701412E 04	0.44744230E 01
511	121	0.28913760E 01	0.50863025E 01	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
517	127	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38	0.00000000E-38
523	133	0.00000000E-38	0.00000000E-38	0.00000000E-38			

```
TIME FROM JD 2433282.5 =  0.55950000E 04   DAYS          ,   0.55204885E 05   SEC (U.T.)
R VECTOR (1950)        =  0.50591229E 04   0.14618619E 04   -0.52419600E 04
V VECTOR (1950)        =  0.43537152E 01   0.28569716E 01    0.52145808E 01
R VECTOR (DATE)        =  0.50618230E 04   0.14788571E 04   -0.52345809E 04
V VECTOR (DATE)        =  0.43365432E 01   0.28714437E 01    0.52209366E 01

TIME FROM JD 2433282.5 =  0.55950000E 04   DAYS          ,   0.55229771E 05   SEC (U.T.)
R VECTOR (1950)        =  0.51659381E 04   0.15325139E 04   -0.51106241E 04
V VECTOR (1950)        =  0.42301896E 01   0.28207926E 01    0.53399309E 01
R VECTOR (DATE)        =  0.51682101E 04   0.15498641E 04   -0.51030891E 04
V VECTOR (DATE)        =  0.42129571E 01   0.28348498E 01    0.53461065E 01

TIME FROM JD 2433282.5 =  0.55950000E 04   DAYS          ,   0.55254657E 05   SEC (U.T.)
R VECTOR (1950)        =  0.52696448E 04   0.16022436E 04   -0.49762064E 04
V VECTOR (1950)        =  0.41039126E 01   0.27828552E 01    0.54622651E 01
R VECTOR (DATE)        =  0.52714873E 04   0.16199384E 04   -0.49685200E 04
V VECTOR (DATE)        =  0.40866299E 01   0.27964883E 01    0.54682565E 01

TIME FROM JD 2433282.5 =  0.55950000E 04   DAYS          ,   0.55279542E 05   SEC (U.T.)
R VECTOR (1950)        =  0.53701755E 04   0.16710075E 04   -0.48387824E 04
V VECTOR (1950)        =  0.39749546E 01   0.27431776E 01    0.55814966E 01
R VECTOR (DATE)        =  0.53715873E 04   0.16890362E 04   -0.48309492E 04
V VECTOR (DATE)        =  0.39576321E 01   0.27563777E 01    0.55872999E 01

TIME FROM JD 2433282.5 =  0.55950000E 04   DAYS          ,   0.55304429E 05   SEC (U.T.)
R VECTOR (1950)        =  0.54674644E 04   0.17387626E 04   -0.46984309E 04
V VECTOR (1950)        =  0.38433883E 01   0.27017791E 01    0.56975399E 01
R VECTOR (DATE)        =  0.54684448E 04   0.17571143E 04   -0.46904557E 04
V VECTOR (DATE)        =  0.38260364E 01   0.27145376E 01    0.57031512E 01

TIME FROM JD 2433282.5 =  0.55950000E 04   DAYS          ,   0.55329314E 05   SEC (U.T.)
R VECTOR (1950)        =  0.55614473E 04   0.18054662E 04   -0.45552320E 04
V VECTOR (1950)        =  0.37092887E 01   0.26586806E 01    0.58103111E 01
R VECTOR (DATE)        =  0.55619955E 04   0.18241298E 04   -0.45471195E 04
V VECTOR (DATE)        =  0.36919178E 01   0.26709890E 01    0.58157269E 01
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,    0.55354200E 05   SEC (U.T.)
R VECTOR (1950)        =   0.56520623E 04   0.18710763E 04   -0.44092683E 04
V VECTOR (1950)        =   0.35727330E 01   0.26139039E 01    0.59197282E 01
R VECTOR (DATE)        =   0.56521782E 04   0.18900406E 04   -0.44010236E 04
V VECTOR (DATE)        =   0.35553537E 01   0.26257541E 01    0.59249448E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,    0.55379086E 05   SEC (U.T.)
R VECTOR (1950)        =   0.57392494E 04   0.19355515E 04   -0.42606243E 04
V VECTOR (1950)        =   0.34338008E 01   0.25674726E 01    0.60257108E 01
R VECTOR (DATE)        =   0.57389328E 04   0.19548049E 04   -0.42522522E 04
V VECTOR (DATE)        =   0.34164236E 01   0.25788567E 01    0.60307247E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,    0.55403972E 05   SEC (U.T.)
R VECTOR (1950)        =   0.58229503E 04   0.19988509E 04   -0.41093862E 04
V VECTOR (1950)        =   0.32925738E 01   0.25194113E 01    0.61281807E 01
R VECTOR (DATE)        =   0.58222014E 04   0.20183817E 04   -0.41008919E 04
V VECTOR (DATE)        =   0.32752093E 01   0.25303217E 01    0.61329886E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,    0.55428858E 05   SEC (U.T.)
R VECTOR (1950)        =   0.59031088E 04   0.20609341E 04   -0.39556426E 04
V VECTOR (1950)        =   0.31491355E 01   0.24697460E 01    0.62270615E 01
R VECTOR (DATE)        =   0.59019280E 04   0.20807305E 04   -0.39470313E 04
V VECTOR (DATE)        =   0.31317943E 01   0.24801752E 01    0.62316602E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,    0.55453744E 05   SEC (U.T.)
R VECTOR (1950)        =   0.59796711E 04   0.21217618E 04   -0.37994837E 04
V VECTOR (1950)        =   0.30035721E 01   0.24185040E 01    0.63222792E 01
R VECTOR (DATE)        =   0.59780592E 04   0.21418116E 04   -0.37907606E 04
V VECTOR (DATE)        =   0.29862648E 01   0.24284452E 01    0.63266655E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,    0.55478630E 05   SEC (U.T.)
R VECTOR (1950)        =   0.60525854E 04   0.21812949E 04   -0.36410016E 04
V VECTOR (1950)        =   0.28559715E 01   0.23657140E 01    0.64137614E 01
R VECTOR (DATE)        =   0.60505433E 04   0.22015860E 04   -0.36321720E 04
V VECTOR (DATE)        =   0.28387087E 01   0.23751602E 01    0.64179325E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,    0.55503516E 05   SEC (U.T.)
```

```
R VECTOR (1950)            =  0.61218021F 04   0.22394954E 04   -0.34802900E 04
V VECTOR (1950)            =  0.27064234E 01   0.23114056E 01    0.65014388E 01
R VECTOR (DATE)            =  0.61193311E 04   0.22600153E 04   -0.34713593E 04
V VECTOR (DATE)            =  0.26892158E 01   0.23203506E 01    0.65053916E 01

TIME FRCM JD 2433282.5 =  0.55950000E 04   DAYS          ,    0.55528402E 05   SEC (U.T.)
R VECTOR (1950)            =  0.61872740E 04   0.22963257F 04   -0.33174446E 04
V VECTOR (1950)            =  0.25550197E 01   0.22556103E 01    0.65852439E 01
R VECTOR (DATE)            =  0.61843755E 04   0.23170620E 04   -0.33084182E 04
V VECTOR (DATE)            =  0.25378779E 01   0.22640478E 01    0.65889759E 01

TIME FRCM JD 2433282.5 =  0.55950000E 04   DAYS          ,    0.55553288E 05   SEC (U.T.)
R VECTOR (1950)            =  0.62489559E 04   0.23517494E 04   -0.31525625E 04
V VECTOR (1950)            =  0.24018543E 01   0.21983604E 01    0.66651118E 01
R VECTOR (DATE)            =  0.62456317F 04   0.23726892F 04   -0.31434460E 04
V VECTOR (DATE)            =  0.23847889E 01   0.22062846E 01    0.66686203E 01

TIME FRCM JD 2433282.5 =  0.55950000E 04   DAYS          ,    0.55578174E 05   SEC (U.T.)
R VECTOR (1950)            =  0.63068051E 04   0.24057306E 04   -0.29857423F 04
V VECTOR (1950)            =  0.22470223E 01   0.21396894E 01    0.67409802E 01
R VECTOR (DATE)            =  0.63030573E 04   0.24268611E 04   -0.29765413E 04
V VECTOR (DATE)            =  0.22300439E 01   0.21470948E 01    0.67442628E 01

TIME FROM JD 2433282.5 =  0.55950000E 04   DAYS          ,    0.55603060E 05   SEC (U.T.)
R VECTOR (1950)            =  0.63607814E 04   0.24582343E 04   -0.28170845E 04
V VECTOR (1950)            =  0.20906211E 01   0.20796323E 01    0.68127895E 01
R VECTOR (DATE)            =  0.63566123E 04   0.24795427E 04   -0.28078047E 04
V VECTOR (DATE)            =  0.20737402E 01   0.20865138E 01    0.68158439E 01

TIME FROM JD 2433282.5 =  0.55950000E 04   DAYS          ,    0.55627946E 05   SEC (U.T.)
R VECTOR (1950)            =  0.64108470E 04   0.25092266E 04   -0.26466907E 04
V VECTOR (1950)            =  0.19327496E 01   0.20182253E 01    0.68804826E 01
R VECTOR (DATE)            =  0.64062592E 04   0.25306997E 04   -0.26373378E 04
V VECTOR (DATE)            =  0.19159767E 01   0.20245779E 01    0.68833067E 01

TIME FROM JD 2433282.5 =  0.55950000E 04   DAYS          ,    0.55652832E 05   SEC (U.T.)
R VECTOR (1950)            =  0.64569668E 04   0.25586744E 04   -0.24746643E 04
```

```
V VECTOR (1950)          =   0.17735081E 01   0.19555055E 01    0.69440053E 01
R VECTOR (DATE)          =   0.64519630E 04   0.25802989E 04   -0.24652439E 04
V VECTOR (DATE)          =   0.17568537E 01   0.19613248E 01    0.69465971E 01

TIME FROM JD 2433282.5 =     0.55950000E 04   DAYS          ,   0.55677718E 05   SEC (U.T.)
R VECTOR (1950)          =   0.64991075E 04   0.26065453E 04   -0.23011093E 04
V VECTOR (1950)          =   0.16129985E 01   0.18915113E 01    0.70033065E 01
R VECTOR (DATE)          =   0.64936909E 04   0.26233080E 04   -0.22916274E 04
V VECTOR (DATE)          =   0.15964730E 01   0.18967930E 01    0.70056641E 01

TIME FROM JD 2433282.5 =     0.55950000E 04   DAYS          ,   0.55702604E 05   SEC (U.T.)
R VECTOR (1950)          =   0.65372392E 04   0.26528083E 04   -0.21261316E 04
V VECTOR (1950)          =   0.14513241E 01   0.18262822E 01    0.70583377E 01
R VECTOR (DATE)          =   0.65314130E 04   0.26746957E 04   -0.21165939E 04
V VECTOR (DATE)          =   0.14349379E 01   C.18310226E 01    0.70604594E 01

TIME FROM JD 2433282.5 =     0.55950000E 04   DAYS          ,   0.55727490E 05   SEC (U.T.)
R VECTOR (1950)          =   0.65713339E 04   0.26974330E 04   -0.19498380E 04
V VECTOR (1950)          =   0.12885897E 01   0.17598590E 01    0.71090534E 01
R VECTOR (DATE)          =   0.65651017E 04   0.27194316E 04   -0.19402505E 04
V VECTOR (DATE)          =   0.12723530E 01   0.17640546E 01    0.71109378E 01

TIME FROM JD 2433282.5 =     0.55950000E 04   DAYS          ,   0.55752376E 05   SEC (U.T.)
R VECTOR (1950)          =   0.66013668E 04   0.27403903E 04   -0.17723365E 04
V VECTOR (1950)          =   0.11249010E 01   0.16922832E 01    0.71554115E 01
R VECTOR (DATE)          =   0.65947325E 04   0.27624865E 04   -0.17627050E 04
V VECTOR (DATE)          =   0.11088240E 01   0.16959308E 01    0.71570570E 01

TIME FROM JD 2433282.5 =     0.55950000E 04   DAYS          ,   0.55777262E 05   SEC (U.T.)
R VECTOR (1950)          =   0.66273153E 04   0.27816520E 04   -0.15937358E 04
V VECTOR (1950)          =   0.96036485E 00   0.16235976E 01    0.71973729E 01
R VECTOR (DATE)          =   0.66202830E C4   0.28038321E 04   -0.15840664E 04
V VECTOR (DATE)          =   0.94445777E 00   0.16266945E 01    0.71987783E 01

TIME FROM JD 2433282.5 =     C.55950000E 04   DAYS          ,   0.55802148E 05   SEC (U.T.)
R VECTOR (1950)          =   0.66491599E 04   0.28211911E 04   -0.14141461E 04
V VECTOR (1950)          =   0.79508959E 00   0.15538460E 01    0.72349015E 01
```

```
R VECTOR (DATE)            =  0.66417339E 04   0.28434414E 04  -0.14044447E 04
V VECTOR (DATE)            =  0.77936238E 00   0.15563899E 01   0.72360658E 01

TIME FROM JD 2433282.5 =  0.55950000E 04   DAYS         ,   0.55827034E 05   SEC (U.T.)
R VECTOR (1950)            =  0.66668834E 04   0.28589815E 04  -0.12336779E 04
V VECTOR (1950)            =  0.62918416E 00   0.14830732E 01   0.72679647E 01
R VECTOR (DATE)            =  0.66590684E 04   0.28812882E 04  -0.12239506E 04
V VECTOR (DATE)            =  0.61364674E 00   0.14850619E 01   0.72688870E 01

TIME FROM JD 2433282.5 =  0.55950000E 04   DAYS         ,   0.55851920E 05   SEC (U.T.)
R VECTOR (1950)            =  0.66804713E 04   0.28949984E 04  -0.10524427E 04
V VECTOR (1950)            =  0.46275826E 00   0.14113247E 01   0.72965332E 01
R VECTOR (DATE)            =  0.66722722E 04   0.29173477E 04  -0.10426955E 04
V VECTOR (DATE)            =  0.44742044E 00   0.14127566E 01   0.72972127E 01

TIME FROM JD 2433282.5 =  0.55950000E 04   DAYS         ,   0.55876806E 05   SEC (U.T.)
R VECTOR (1950)            =  0.66899124E 04   0.29292181E 04  -0.87055280E 03
V VECTOR (1950)            =  0.29592245E 00   0.13386469E 01   0.73205808E 01
R VECTOR (DATE)            =  0.66813340E 04   0.29515961E 04  -0.86079163E 03
V VECTOR (DATE)            =  0.28079392E 00   0.13395207E 01   0.73210168E 01

TIME FROM JD 2433282.5 =  0.55950000E 04   DAYS         ,   0.55901691E 05   SEC (U.T.)
R VECTOR (1950)            =  0.66951974E 04   0.29616181E 04  -0.68812098E 03
V VECTOR (1950)            =  0.12878812E 00   0.12650874E 01   0.73400851E 01
R VECTOR (DATE)            =  0.66862454E 04   0.29840109E 04  -0.67835200E 03
V VECTOR (DATE)            =  0.11387843E 00   0.12654022E 01   0.73402773E 01

TIME FROM JD 2433282.5 =  0.55950000E 04   DAYS         ,   0.55926578E 05   SEC (U.T.)
R VECTOR (1950)            =  0.66963208E 04   0.29921771E 04  -0.50526051E 03
V VECTOR (1950)            = -0.38532974E-01   0.11906944E 01   0.73550269E 01
R VECTOR (DATE)            =  0.66870005E 04   0.30145707E 04  -0.49548978E 03
V VECTOR (DATE)            = -0.53214395E-01   0.11904496E 01   0.73549749E 01

TIME FROM JD 2433282.5 =  0.55950000E 04   DAYS         ,   0.55951463E 05   SEC (U.T.)
R VECTOR (1950)            =  0.66932790E 04   0.30208748E 04  -0.32208517E 03
V VECTOR (1950)            = -0.20592881E 00   0.11155168E 01   0.73653905E 01
R VECTOR (DATE)            =  0.66835963E 04   0.30432553E 04  -0.31231878E 03
```

```
V VECTOR (DATE)         =  -0.22037268E 0C   0.11147122E 01   0.73650942E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS         ,   0.55976350E 05   SEC (U.T.)
R VECTOR (1950)        =   0.66860717E 04   0.30476924E 04  -0.13870889E 03
V VECTOR (1950)        =  -0.37328678E 00   0.10396042E 01   0.73711637E 01
R VECTOR (DATE)        =   0.66760325E 04   0.30700460E 04  -0.12895291E 03
V VECTOR (DATE)        =  -0.38748395E 00   0.10382400E 01   0.73706232E 01

TIME FROM JD 2433282.5 =   0.5595[illegible]0E 04   DAYS         ,   0.56001235E 05   SEC (U.T.)
R VECTOR (1950)        =   0.66747010E 04   0.30726122E 04   0.44753768E 02
V VECTOR (1950)        =  -0.54049421E 00   0.96300703E 00   0.73723379E 01
R VECTOR (DATE)        =   0.66643118E 04   0.30949248E 04   0.54493260E 02
V VECTOR (DATE)        =  -0.55443572E 00   0.96108390E 00   0.73715534E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS         ,   0.56026122E 05   SEC (U.T.)
R VECTOR (1950)        =   0.66591724E 04   0.30956178E 04   0.22818842E 03
V VECTOR (1950)        =  -0.70743819E 00   0.88577627E 00   0.73689080E 01
R VECTOR (DATE)        =   0.66484394E 04   0.31178757E 04   0.23790536E 03
V VECTOR (DATE)        =  -0.72111521E 00   0.88329510E 00   0.73678799E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS         ,   0.56051007E 05   SEC (U.T.)
R VECTOR (1950)        =   0.66394937E 04   0.31166942E 04   0.41148028E 03
V VECTOR (1950)        =  -0.87400591E 00   0.80796351F 00   0.73608726E 01
R VECTOR (DATE)        =   0.66284237E 04   0.31388834E 04   0.42116861E 03
V VECTOR (DATE)        =  -0.88740981E 00   0.80492564F 00   0.73596014E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS         ,   0.56075894E 05   SEC (U.T.)
R VECTOR (1950)        =   0.66156755E 04   0.31358273E 04   0.59451501E 03
V VECTOR (1950)        =  -0.10400850F 01   0.72962068F 00   0.73482336E 01
R VECTOR (DATE)        =   0.66042755E 04   0.31579340E 04   0.60416869E 03
V VECTOR (DATE)        =  -0.10532073E 01   0.72602780E 00   0.73467202E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS         ,   0.56100779E 05   SEC (U.T.)
R VECTOR (1950)        =   0.65877315E 04   0.31530048E 04   0.77717802E 03
V VECTOR (1950)        =  -0.12055628E 01   0.65080047E 00   0.73309969E 01
R VECTOR (DATE)        =   0.65760085E 04   0.31750152E 04   0.78679103E 03
V VECTOR (DATE)        =  -0.12183952E 01   0.64665467E 00   0.73292419E 01
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS        ,   0.56125666E 05   SEC (U.T.)
R VECTOR (1950)        =   0.65556782E 04   0.31682154E 04   0.95935477E 03
V VECTOR (1950)        =  -0.13703271E 01   0.57155591E 00   0.73091718E 01
R VECTOR (DATE)        =   0.65436396E 04   0.31901158E 04   0.96892111E 03
V VECTOR (DATE)        =  -0.13828616E 01   0.56685966E 00   0.73071765E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS        ,   0.56150551E 05   SEC (U.T.)
R VECTOR (1950)        =   0.65195346E 04   0.31814492E 04   0.11409315E 04
V VECTOR (1950)        =  -0.15342662E 01   0.49194038E 00   0.72827708E 01
R VECTOR (DATE)        =   0.65071878E 04   0.32032259E 04   0.11504452E 04
V VECTOR (DATE)        =  -0.15464948E 01   0.48669649E 00   0.72805362E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS        ,   0.56175437E 05   SEC (U.T.)
R VECTOR (1950)        =   0.64793226E 04   0.31926976E 04   0.13217945E 04
V VECTOR (1950)        =  -0.16972697E 01   0.41200733E 00   0.72518108E 01
R VECTOR (DATE)        =   0.64666754E 04   0.32143369E 04   0.13312496E 04
V VECTOR (DATE)        =  -0.17091848E 01   0.40621902E 00   0.72493384E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS        ,   0.56200323E 05   SEC (U.T.)
R VECTOR (1950)        =   0.64350668E 04   0.32019534E 04   0.15018305E 04
V VECTOR (1950)        =  -0.18592268E 01   0.33181091E 00   0.72163118E 01
R VECTOR (DATE)        =   0.64221271E 04   0.32234420E 04   0.15112211E 04
V VECTOR (DATE)        =  -0.18708209E 01   0.32548174E 00   0.72136030E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS        ,   0.56225209E 05   SEC (U.T.)
R VECTOR (1950)        =   0.63867949E 04   0.32092107E 04   0.16809268E 04
V VECTOR (1950)        =  -0.20200276E 01   0.25140537E 00   0.71762971E 01
R VECTOR (DATE)        =   0.63735706E 04   0.32305351E 04   0.16902471E 04
V VECTOR (DATE)        =  -0.20312934E 01   0.24453928E 00   0.71733536E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS        ,   0.56250095E 05   SEC (U.T.)
R VECTOR (1950)        =   0.63345367E 04   0.32144651E 04   0.18589713E 04
V VECTOR (1950)        =  -0.21795640E 01   0.17084487E 00   0.71317939E 01
R VECTOR (DATE)        =   0.63210363E 04   0.32356119E 04   0.18682155E 04
V VECTOR (DATE)        =  -0.21904945E 01   0.16344615E 00   0.71286176E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS            0.56274981E 05   SEC (U.T.)
```

```
R VECTOR (1950)          =  0.62783251E 04   0.32177131E 04   0.20358529E 04
V VECTOR (1950)          = -0.23377284E 01   0.90183808E-01   0.70828327E 01
R VECTOR (DATE)          =  0.62645569E 04   0.32386693E 04   0.20450151F 04
V VECTOR (DATE)          = -0.23483167E 01   0.82257124E-01   0.70794256E 01

  FLOYD  OBSERVES
    RELATIVE POSITION    =  0.17567471E 04   0.20687971E 04  -0.22986172E 04
    RELATIVE VELOCITY    = -0.22524199E 01  -0.23853086E 00   0.70828327E 01
    R, RDOT, AZ, ELEV    =  0.35566449E 04  -0.58288431E 01   0.15396125E 03   0.72543477E 00

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0    ) =   0.55950000E 04   0.72833081E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =   0.56274981E 05   0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )  =   0.61279427E 04   0.31832442E 04   0.21339417E 04
VELOCITY VECTOR ( DATE )  =  -0.25601562E 01  -0.26524332E-01   0.70425999E 01
DELTA RADIUS              =  -0.13661419E 03  -0.55425065E 02   0.88926515E 02
DELTA VELOCITY            =  -0.21183954E 00  -0.10878146E 00  -0.36825711E-01
```

THE COVARIANCE MATRIX FOR THE ERRORS IN STATE ** P **

0.21330771E 03	0.86258942E 02	-0.13872655E 03	0.33055931E 00	0.16980946E 00	0.57473525E-01
0.86258942E 02	0.35277802E 02	-0.56293343E 02	0.13400103E 00	0.68687804E-01	0.23348182E-01
-0.13872655E 03	-0.56293350E 02	0.90331177E 02	-0.21513200E 00	-0.11043151E 00	-0.37412979E-01
0.33055931E 00	0.13400103E 00	-0.21513200E 00	0.51258923E-03	0.26316513E-03	0.89253357E-04
0.16980943E 00	0.68687797E-01	-0.11043149E 00	0.26316513E-03	0.13530106E-03	0.45844550E-04
0.57473533E-01	0.23348186E-01	-0.37412994E-01	0.89253372E-04	0.45844565E-04	0.15768717E-04

THERE WAS NO RECTIFICATION AT THIS TIME

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,   0.56284978E 05   SEC (U.T.)
R VECTOR (1950)        =   0.62546403E 04   0.32184527E 04    0.21065527E 04
V VECTOR (1950)        =  -0.24008514E 01   0.57766852E-01    0.70619178E 01
R VECTOR (DATE)        =   0.62407670E 04   0.32393286E 04    0.21156804E 04
V VECTOR (DATE)        =  -0.24113004E 01   0.49629479E-01    0.70584186E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.17338110E 04   0.20661719E 04   -0.22279520E 04
    RELATIVE VELOCITY  =  -0.23153034E 01  -0.27088556E 00    0.70619178E 01
    R, RDOT, AZ, ELEV  =   0.34984192E 04  -0.58047956E 01    0.15346003E 03   0.12704640E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,   0.56294974E 05   SEC (U.T.)
R VECTOR (1950)        =   0.62303258E 04   0.32188681E 04    0.21770396E 04
V VECTOR (1950)        =  -0.24637286E 01   0.25346192E-01    0.70402915E 01
R VECTOR (DATE)        =   0.62163487E 04   0.32396616E 04    0.21861319E 04
V VECTOR (DATE)        =  -0.24740373E 01   0.16998963E-01    0.70367005E 01

  FLCYD  OBSERVES
    RELATIVE POSITION  =   0.17102487E 04   0.20632215E 04   -0.21575005E 04
    RELATIVE VELOCITY  =  -0.23779413E 01  -0.30324380E 00    0.70402915E 01
    R, RDOT, AZ, ELEV  =   0.34404421E 04  -0.57788975E 01    0.15294119E 03   0.18231791E 01

UNDRFLOW AT 33550 IN MQ
```

TIME FROM JD 2433282.5 =	0.55950000E 04	DAYS ,	0.56284978E 05	SEC (U.T.)
R VECTOR (1950) =	0.62544403E 04	0.32184527E 04	0.21065527E 04	
V VECTOR (1950) =	-0.24008514E 01	0.57766852E-01	0.70619178E 01	
R VECTOR (DATE) =	0.62407670E 04	0.32393286E 04	0.21156804E 04	
V VECTOR (DATE) =	-0.24113004E 01	0.49629479E-01	0.70584186E 01	

FLOYD OBSERVES

RELATIVE POSITION =	0.17338110E 04	0.20661719E 04	-0.22279520E 04	
RELATIVE VELOCITY =	-0.23153034E 01	-0.27088556E 00	0.70619178E 01	
R, RDOT, AZ, ELEV =	0.34984192E 04	-0.58047956E 01	0.15346003E 03	0.12704640E 01

TIME FROM JD 2433282.5 =	0.55950000E 04	DAYS ,	0.56294974E 05	SEC (U.T.)
R VECTOR (1950) =	0.62303258E 04	0.32188681E 04	0.21770396E 04	
V VECTOR (1950) =	-0.24637286E 01	0.25346192E-01	0.70402915E 01	
R VECTOR (DATE) =	0.62163487E 04	0.32396616E 04	0.21861319E 04	
V VECTOR (DATE) =	-0.24740373E 01	0.16998963E-01	0.70367005E 01	

FLOYD OBSERVES

RELATIVE POSITION =	0.17102487E 04	0.20632215E 04	-0.21575005E 04	
RELATIVE VELOCITY =	-0.23779413E 01	-0.30324380E 00	0.70402915E 01	
R, RDOT, AZ, ELEV =	0.34404421E 04	-0.57788975E 01	0.15294119E 03	0.18231791E 01

UNDRFLOW AT 33550 IN MQ

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS           ,   0.56304985E 05   SEC (U.T.)
R VECTOR (1950)        =   0.62053464E 04   0.32189593E 04     0.22474107E 04
V VECTOR (1950)        =  -0.25264467E 01  -0.71231084E-02     0.70179226E 01
R VECTOR (DATE)        =   0.61912668E 04   0.32396682E 04     0.22564666E 04
V VECTOR (DATE)        =  -0.25366138E 01  -0.15679650E-01     0.70142401E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.16860276E 04   0.20599353E 04    -0.20871658E 04
    RELATIVE VELOCITY  =  -0.24404192E 01  -0.33565032E 00     0.70179226E 01
    R, RDOT, AZ, ELEV  =   0.33826445E 04  -0.57510036E 01     0.15240308E 03   0.23846964E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS           ,   0.56314997E 05   SEC (U.T.)
R VECTOR (1950)        =   0.61797406E 04   0.32187255E 04     0.23175542E 04
V VECTOR (1950)        =  -0.25889044E 01  -0.39588966E-01     0.69948452E 01
R VECTOR (DATE)        =   0.61655599E 04   0.32393477E 04     0.23265728E 04
V VECTOR (DATE)        =  -0.25989289E 01  -0.48353938E-01     0.69910715E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.16611828E C4   0.20563277E 04    -0.20170596E 04
    RELATIVE VELOCITY  =  -0.25026371E 01  -0.36805332E 00     0.69948452E 01
    R, RDOT, AZ, ELEV  =   0.33251378E 04  -0.57210248E 01     0.15184521E 03   0.29544208E 01

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0    ) =   0.55950000E 04   0.72879395E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =   0.56314997E 05   0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )  =   0.60145764E 04   0.31770894E 04   0.24173471E 04
VELOCITY VECTOR ( DATE )  =  -0.28258927E 01  -0.16560688E 00   0.69446530E 01
DELTA RADIUS              =  -0.15098346E 03  -0.62258314E 02   0.90774340E 02
DELTA VELOCITY            =  -0.22696388E 00  -0.11725294E 00  -0.46418507E-01
```

THE COVARIANCE MATRIX FOR THE ERRORS IN STATE ** P **

0.85230084E 02	0.34854742E 02	-0.51125036E 02	0.12791006E 00	0.66149730E-01	0.26159544E-01
0.34854739E 02	0.14639354E 02	-0.21086090E 02	0.52622620E-01	0.27055835E-01	0.10812853E-01
-0.51125029E 02	-0.21086097E 02	0.30766915E 02	-0.76857375E-01	-0.39661304E-01	-0.15714646E-01
0.12791008E 00	0.52622634E-01	-0.76857400E-01	0.19227092E-03	0.99272188E-04	0.39445080E-04
0.66149705E-01	0.27055832E-01	-0.39661298E-01	0.99272181E-04	0.51460083E-04	0.20385964E-04
0.26159552E-01	0.10812859E-01	-0.15714664E-01	0.39445089E-04	0.20385980E-04	0.83130838E-05

THERE WAS NO RECTIFICATION AT THIS TIME

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS         ,    0.56324987E 05   SEC (U.T.)
R VECTOR (1950)        =   0.61535655E 04   0.32181682E 04    0.23873195E 04
V VECTOR (1950)        =  -0.26509676E 01  -0.71981271E-01    0.69711114E 01
R VECTOR (DATE)        =   0.61392854E 04   0.32387018E 04    0.23963000E 04
V VECTOR (DATE)        =  -0.26608488E 01  -0.80953345E-01    0.69672471E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.16357715E 04   0.20523999E 04   -0.19473324E 04
    RELATIVE VELOCITY  =  -0.25644611E 01  -0.40038268E 00    0.69711114E 01
    R, RDOT, AZ, ELEV  =   0.32680571E 04  -0.56889117E 01    0.15126789E 03   0.35314470E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS         ,    0.56334978E 05   SEC (U.T.)
R VECTOR (1950)        =   0.61267715E 04   0.32172872E 04    0.24568442E 04
V VECTOR (1950)        =  -0.27127592E 01  -0.10436357E 00    0.69466773E 01
R VECTOR (DATE)        =   0.61123934E 04   0.32377301E 04    0.24657856E 04
V VECTOR (DATE)        =  -0.27224961E 01  -0.11354183E 00    0.69427229E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.16097451E 04   0.20481473E 04   -0.18778468E 04
    RELATIVE VELOCITY  =  -0.26260134E 01  -0.43270186E 00    0.69466773E 01
    R, RCOT, AZ, ELEV  =   0.32113073E 04  -0.56544712E 01    0.15066878E 03   0.41171793E 01

UNDRFLOW AT 33550 IN MQ
```


203-1

```
EPOCH (REL. TO 1950.0     ) =   0.55950000E 04   0.72902521E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =     0.56334978E 05   0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )  =     0.59558820E 04   0.31726944E 04   0.25561499E 04
VELOCITY VECTOR ( DATE )  =    -0.29541987E 01  -0.23362684E 00   0.68916765E 01
DELTA RADIUS              =    -0.15651134E 03  -0.65035681E 02   0.90364326E 02
DELTA VELOCITY            =    -0.23170260E 00  -0.12008501E 00  -0.51046385E-01
```

THE COVARIANCE MATRIX FOR THE ERRORS IN STATE ** P **

0.69199964E 02	0.28459142E 02	-0.39839622E 02	0.10222960E 00	0.53054717E-01	0.22515357E-01
0.28459137E 02	0.12084222E 02	-0.16555414E 02	0.42350352E-01	0.21816103E-01	0.93783418E-02
-0.39839614E 02	-0.16555421E 02	0.23032245E 02	-0.58977371E-01	-0.30520273E-01	-0.12979564E-01
0.10222962E 00	0.42350369E-01	-0.58977400E-01	0.15132428E-03	0.78367439E-04	0.33447122E-04
0.53054692E-01	0.21816101E-01	-0.30520268E-01	0.78367428E-04	0.40796138E-04	0.17340685E-04
0.22515366E-01	0.93783485E-02	-0.12979582E-01	0.33447130E-04	0.17340701E-04	0.76099270E-05

THERE WAS NO RECTIFICATION AT THIS TIME

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,   0.56344969E 05   SEC (U.T.)
R VECTOR (1950)        =   0.60993615E 04   0.32160828E 04    0.25261216E 04
V VECTOR (1950)        =  -0.27742719E 01  -0.13673216E 00    0.69215456E 01
R VECTOR (DATE)        =   0.60848868E 04   0.32364330E 04    0.25350230E 04
V VECTOR (DATE)        =  -0.27838635E 01  -0.14611566E 00    0.69175014E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.15831064E 04   0.20435696E 04   -0.18086094E 04
    RELATIVE VELOCITY  =  -0.26872869E 01  -0.46500716E 00    0.69215456E 01
    R, RDOT, AZ, ELEV  =   0.31549121E 04  -0.56175600E 01    0.15004668E 03   0.47118833E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,   0.56354959E 05   SEC (U.T.)
R VECTOR (1950)        =   0.60713386E 04   0.32145552E 04    0.25951442E 04
V VECTOR (1950)        =  -0.28354985E 01  -0.16908324E 00    0.68957193E 01
R VECTOR (DATE)        =   0.60567689E 04   0.32348106E 04    0.26040047E 04
V VECTOR (DATE)        =  -0.28449440E 01  -0.17867101E 00    0.68915858E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.15558588E 04   0.20386675E 04   -0.17396277E 04
    RELATIVE VELOCITY  =  -0.27482744E 01  -0.49729477E 00    0.68957193E 01
    R, RDOT, AZ, ELEV  =   0.30988976E 04  -0.55780255E 01    0.14940028E 03   0.53158023E 01

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0     ) =     0.55950000E 04   0.72925648E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =      0.56354959E 05   0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )   =     0.58946159E 04   0.31669266E 04   0.26938536E 04
VELOCITY VECTOR ( DATE )   =    -0.30813822E 01  -0.30161882E 00   0.68356852E 01
DELTA RADIUS               =    -0.16215291E 03  -0.67883995E 02   0.89848964E 02
DELTA VELOCITY             =    -0.23643826E 00  -0.12294781E 00  -0.55900542E-01
```

THE COVARIANCE MATRIX FOR THE ERRORS IN STATE ** P **

0.60065925E 02	0.24843406E 02	-0.33171648E 02	0.87362177E-01	0.45502786E-01	0.20640823E-01
0.24843400E 02	0.10649808E 02	-0.13883401E 02	0.36434189E-01	0.18810024E-01	0.86592562E-02
-0.33171639E 02	-0.13883409E 02	0.18411798E 02	-0.48359526E-01	-0.25099399E-01	-0.11410224E-01
0.87362201E-01	0.36434209E-01	-0.48359657E-01	0.12735447E-03	0.66162875E-04	0.30205076E-04
0.45502761E-01	0.18810022E-01	-0.25099396E-01	0.66162861E-04	0.34591323E-04	0.15710564E-04
0.20640832E-01	0.86592638E-02	-0.11410242E-01	0.30205082E-04	0.15710581E-04	0.73771830E-05

THERE WAS NO RECTIFICATION AT THIS TIME

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS           ,    0.56364971E 05   SEC (U.T.)
R VECTOR (1950)        =   0.60426458E 04   0.32127003E 04    0.26640467E 04
V VECTOR (1950)        =  -0.28965586E 01  -0.20148027E 00    0.68691460E 01
R VECTOR (DATE)        =   0.60279822E 04   0.32328587E 04    0.26728654E 04
V VECTOR (DATE)        =  -0.29058565E 01  -0.21127174E 00    0.68649234E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.15279464E 04   0.20334314E 04   -0.16707670E 04
    RELATIVE VELOCITY  =  -0.28090949E 01  -0.52962805E 00    0.68691460E 01
    R, RDOT, AZ, ELEV  =   0.30431769E 04  -0.55356134E 01    0.14872675E 03   0.59304416E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS           ,    0.56374982E 05   SEC (U.T.)
R VECTOR (1950)        =   0.60133430E 04   0.32105211E 04    0.27326799E 04
V VECTOR (1950)        =  -0.29573191E 01  -0.23385306E 00    0.68418812E 01
R VECTOR (DATE)        =   0.59985871E 04   0.32305805E 04    0.27414559E 04
V VECTOR (DATE)        =  -0.29664687E 01  -0.24384722E 00    0.68375698E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.14994276E 04   0.20278699E 04   -0.16021765E 04
    RELATIVE VELOCITY  =  -0.28696160E 01  -0.56193694E 00    0.68418812E 01
    R, RDOT, AZ, ELEV  =   0.29878937E 04  -0.54902279E 01    0.14802596E 03   0.65548010E 01

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0    ) =   0.55950000E 04   0.72948822E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =    0.56374982E 05   0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )  =    0.58306170E 04   0.31597509E 04   0.28307033E 04
VELOCITY VECTOR ( DATE )  =   -0.32077139E 01  -0.36973028E 00   0.67765544E 01
DELTA RADIUS              =   -0.16797006E 03  -0.70829586E 02   0.89247479E 02
DELTA VELOCITY            =   -0.24124520E 00  -0.12588305E 00  -0.61015408E-01
```

THE COVARIANCE MATRIX FOR THE ERRORS IN STATE ** P **

0.54546620E 02	0.22690979E 02	-0.28875290E 02	0.78114581F-01	0.40837367E-01	0.19734481E-01
0.22690971E 02	0.98079886E 01	-0.12168074E 02	0.32789609E-01	0.16971367E-01	0.83347703E-02
-0.28875280E 02	-0.12168082E 02	0.15375571E 02	-0.41456757E-01	-0.21583229E-01	-0.10452227E-01
0.78114608E-01	0.32789631E-01	-0.41456791E-01	0.11214938E-03	0.58456808E-04	0.28444745E-04
0.40837342E-01	0.16971366E-01	-0.21583226E-01	0.58456792E-04	0.30695965E-04	0.14844458E-04
0.19734490E-01	0.83347792E-02	-0.10452246E-01	0.28444750E-04	0.14844476E-04	0.74239523E-05

THERE WAS NO RECTIFICATION AT THIS TIME

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS         ,   0.56385488E 05   SEC (U.T.)
R VECTOR (1950)        =   0.59819410E 04   0.32078860E 04   0.28044048E 04
V VECTOR (1950)        =  -0.30207498E 01  -0.26779422E 00   0.68125297E 01
R VECTOR (DATE)        =   0.59670897E 04   0.32278393E 04   0.28131351E 04
V VECTOR (DATE)        =  -0.30297428E 01  -0.27799996E 00   0.68081259E 01

  FLOYC  OBSERVES
    RELATIVE POSITION  =   0.14688527E 04   0.20216830E 04  -0.15304973E 04
    RELATIVE VELOCITY  =  -0.29327955E 01  -0.59581083E 00   0.68125297E 01
    R, RDOT, AZ, ELEV  =   0.29303843E 04  -0.54391990E 01   0.14725955E 03   0.72206841E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS         ,   0.56395993E 05   SEC (U.T.)
R VECTOR (1950)        =   0.59498743E C4   0.32048946E 04   0.28758177E 04
V VECTOR (1950)        =  -0.30838359E 01  -0.30170064E 00   0.67824242E 01
R VECTOR (DATE)        =   0.59349294E 04   0.32247395E 04   0.28845012E 04
V VECTOR (DATE)        =  -0.30926712E 01  -0.31211681E 00   0.67779283E 01

  FLCYD  OBSERVES
    RELATIVE POSITION  =   0.14376176E 04   0.20151384E 04  -0.14591312E 04
    RELATIVE VELOCITY  =  -0.29956304E 01  -0.62964979E 00   0.67824242E 01
    R, RDOT, AZ, ELEV  =   0.28734284E 04  -0.53844560E 01   0.14645945E 03   0.78977276E 01

UNDRFLOW AT 33550 IN MQ
```

EPOCH (REL. TO 1950.0)	=	0.55950000E 04	0.72973141E 00	(DAYS,DAYS)
EPOCH (REL. JD 2433282.5)	=	0.56395993E 05	0.55950000E 04	(DAYS U.T.)

POSITICN VECTOR (DATE)	=	0.57606115E 04	0.31506896E 04	0.29730510E 04
VELOCITY VECTOR (DATE)	=	-0.33391123E 01	-0.44119819E 00	0.67112402E 01
DELTA RADIUS	=	-0.17431795E 03	-0.74049925E 02	0.88549795E 02
DELTA VELOCITY	=	-0.24644110E 00	-0.12908138E 00	-0.66688007E-01

THE COVARIANCE MATRIX FOR THE ERRORS IN STATE ** P **

0.51415162E 02	0.21521875E 02	-0.26015666E 02	0.72452914E-01	0.38028637E-01	0.19574591E-01
0.21521865E 02	0.93713008E 01	-0.11038484E 02	0.30615881E-01	0.15895074E-01	0.83221939E-02
-0.26015656E 02	-0.11038492E 02	0.13251286E 02	-0.36757641E-01	-0.19202370E-01	-0.99038760E-02
0.72452944E-01	0.30615905E-01	-0.36757677E-01	0.10237488E-03	0.53556841E-04	0.27766947E-04
0.38028612E-01	0.15895074E-01	-0.19202368E-01	0.53556822E-04	0.28251766E-04	0.14543491E-04
0.19574601E-01	0.83222039E-02	-0.99038952E-02	0.27766950E-04	0.14543511E-04	0.77363826E-05

THERE WAS NO RECTIFICATION AT THIS TIME

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,    0.56405984E 05   SEC (U.T.)
R VECTOR (1950)        =   0.59187661E 04   0.32017194E 04     0.29434330E 04
V VECTOR (1950)        =  -0.31435031E 01  -0.33390931E 00     0.67530980E 01
R VECTOR (DATE)        =   0.59037336E 04   0.32214592E 04     0.29520711E 04
V VECTOR (DATE)        =  -0.31521876E 01  -0.34452448E 00     0.67485150E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.14073043E 04   0.20085813E 04    -0.13915612E 04
    RELATIVE VELOCITY  =  -0.30550587E 01  -0.66179410E 00     0.67530980E 01
    R, RDOT, AZ, ELEV  =   0.28198133E 04  -0.53287265E 01     0.14566545E 03   0.85521011E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,    0.56415975E 05   SEC (U.T.)
R VECTOR (1950)        =   0.58870638E 04   0.31982227E 04     0.30107516E 04
V VECTOR (1950)        =  -0.32028443E 01  -0.36607886E 00     0.67230969E 01
R VECTOR (DATE)        =   0.58719454E 04   0.32178556E 04     0.30193435E 04
V VECTOR (DATE)        =  -0.32113772E 01  -0.37689193E 00     0.67184273E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.13764011E 04   0.20017016E 04    -0.13242888E 04
    RELATIVE VELOCITY  =  -0.31141610E 01  -0.69389912E 00     0.67230969E 01
    R, RDOT, AZ, ELEV  =   0.27667725E 04  -0.52691848E 01     0.14483723E 03   0.92168491E 01

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0    ) =    0.55950000E 04    0.72996268E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =    0.56415975E 05    0.55950000E 04 (DAYS  U.T.)

POSITICN VECTOR ( DATE )  =    0.56905817E 04    0.31402809E 04    0.31075322E 04
VELOCITY VECTOR ( DATE )  =   -0.34639775E 01   -0.50969397E 00    0.66457261E 01
DELTA RADIUS              =   -0.18136372E 03   -0.77574660E 02    0.88188667E 02
DELTA VELOCITY            =   -0.25260032E 00   -0.13280204E 00   -0.72701097E-01


THE COVARIANCE MATRIX FOR THE ERRORS IN STATE  ** P **


 0.49648365E 02   0.20906243E 02  -0.24045115E 02   0.68904984E-01   0.36307040E-01   0.19788938E-01
 0.20906231E 02   0.91596683E 01  -0.10266557E 02   0.29296382E-01   0.15258997E-01   0.84639725E-02
-0.24045103E 02  -0.10266565E 02   0.11731097E 02  -0.33460686E-01  -0.17539524E-01  -0.95774321E-02
 0.68905016E-01   0.29296408E-01  -0.33460723E-01   0.95899636E-04   0.50350723E-04   0.27646667E-04
 0.36307014E-01   0.15258997E-01  -0.17539523E-01   0.50350701E-04   0.26677417E-04   0.14532450E-04
 0.19788949E-01   0.84639837E-02  -0.95774517E-02   0.27646669E-04   0.14532470E-04   0.81712567E-05


THERE WAS NO RECTIFICATION AT THIS TIME
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,    0.56425986E 05   SEC (U.T.)
R VECTOR (1950)        =   0.58547029E 04   0.31943966E 04     0.30779053E 04
V VECTOR (1950)        =  -0.32619762E 01  -0.39827290E 00     0.66923602E 01
R VECTOR (DATE)        =   0.58394998E 04   0.32139202E 04     0.30864500E 04
V VECTOR (DATE)        =  -0.32703563E 01  -0.40928316E 00     0.66876042E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.13448455E 04   0.19944814E 04    -0.12571824E 04
    RELATIVE VELOCITY  =  -0.31730534E 01  -0.72602826E 00     0.66923602E 01
    R, RDOT, AZ, ELEV  =   0.27142352E 04  -0.52054565E 01     0.14397101E 03   0.98934599E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,    0.56435998E 05   SEC (U.T.)
R VECTOR (1950)        =   0.58217518E 04   0.31902484E 04     0.31447478E 04
V VECTOR (1950)        =  -0.33207686E 01  -0.43042098E 00     0.66609523E 01
R VECTOR (DATE)        =   0.58064656E 04   0.32096608E 04     0.31532445E 04
V VECTOR (DATE)        =  -0.33289950E 01  -0.44162729E 00     0.66561106E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.13127021E 04   0.19869427E 04    -0.11903879E 04
    RELATIVE VELOCITY  =  -0.32316066E 01  -0.75811137E 00     0.66609523E 01
    R, RDOT, AZ, ELEV  =   0.26623582E 04  -0.51373919E 01     0.14306610E 03   0.10580531E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0     ) =    0.55950000E 04    0.73019442E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =      0.56435998E 05    0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )  =      0.56175795E 04    0.31283181E 04    0.32410600E 04
VELOCITY VECTOR ( DATE )  =     -0.35881787E 01   -0.57841986E 00    0.65769424E 01
DELTA RADIUS              =     -0.18888611E 03   -0.81342735E 02    0.87815529E 02
DELTA VELOCITY            =     -0.25918372E 00   -0.13679257E 00   -0.79168111E-01
```

THE COVARIANCE MATRIX FOR THE ERRORS IN STATE ** P **

0.49054978E 02	0.20782686E 02	-0.22716513E 02	0.67055902E-01	0.35473424E-01	0.20426592E-0
0.20782673E 02	0.91548818E 01	-0.97586089E 01	0.28684077E-01	0.14993519E-01	0.87874780E-0
-0.22716500E 02	-0.97586169E 01	0.10604215E 02	-0.31134475E-01	-0.16378675E-01	-0.94468949E-0
0.67055937E-01	0.28684105E-01	-0.31134513E-01	0.91924883E-04	0.48446284E-04	0.28104119E-0
0.35473397E-01	0.14993520E-01	-0.16378674E-01	0.48446260E-04	0.25781533E-04	0.14828111E-0
0.20426603E-01	0.87874904E-02	-0.94469149E-02	0.28104119E-04	0.14828123E-04	0.87890600E-0

THERE WAS NO RECTIFICATION AT THIS TIME

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS                 0.56445998E 05   SEC (U.T.)
R VECTOR (1950)        =   0.57882833E 04   0.31857881E 04   0.32111360E 04
V VECTOR (1950)        =  -0.33790959E 01  -0.46245395E 00   0.66289440E 01
R VECTOR (DATE)        =   0.57729156E 04   0.32050876E 04   0.32195839E 04
V VECTOR (DATE)        =  -0.33871681E 01  -0.47385474E 00   0.66240171E 01

  FLOYD  OBSERVES
    RFLATIVE POSITION  =   0.12800443E 04   0.19790954E 04  -0.11240485E 04
    RELATIVE VELOCITY  =  -0.32896951E 01  -0.79007929E 00   0.66289440E 01
    R, RDOT, AZ, ELEV  =   0.26112864E 04  -0.50648799E 01   0.14212242E 03   0.11276614E 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS         ,       0.56455979E 05   SEC (U.T.)
R VECTOR (1950)        =   0.57542341E 04   0.31810082E 04   0.32772009E 04
V VECTOR (1950)        =  -0.34370727E 01  -0.49443427E 00   0.65962749E 01
R VECTOR (DATE)        =   0.57387866E 04   0.32001927E 04   0.32855992F 04
V VECTOR (DATE)        =  -0.34449901E 01  -0.50602836E 00   0.65912634E 01

  FLCYD  OBSERVES
    RELATIVE POSITION  =   0.12468097E 04   0.19709273E 04  -0.10580332E 04
    RELATIVE VELOCITY  =  -0.33474332E 01  -0.82199438E 00   0.65962749E 01
    R, RDOT, AZ, ELEV  =   0.25609613E 04  -0.49874948E 01   0.14113570E 03   0.11982958E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0    ) =    0.55950000E 04   0.73042569E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =    0.56455979E 05   0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )  =    0.55423167E 04   0.31150117E 04   0.33728329E 04
VELOCITY VECTOR ( DATE )  =   -0.37106132E 01  -0.64678333E 00   0.65053384E 01
DELTA RADIUS              =   -0.19646980E 03  -0.85181046E 02   0.87233810E 02
DELTA VELOCITY            =   -0.26562314E 00  -0.14075497E 00  -0.85925034E-01
```

THE COVARIANCE MATRIX FOR THE ERRORS IN STATE ** P **

0.49402808E 02	0.21061005E 02	-0.21853730E 02	0.66521727E-01	0.35333884E-01	0.21447519E-01
0.21060990E 02	0.93222414E 01	-0.94441690E 01	0.28627820E-01	0.15022572E-01	0.92790010E-02
-0.21853717E 02	-0.94441772E 01	0.97509460E 01	-0.29501311E-01	-0.15577873E-01	-0.94694239E-02
0.66521763E-01	0.28627850E-01	-0.29501349E-01	0.89828925E-04	0.47527900E-04	0.29060943E-04
0.35333857E-01	0.15022574E-01	-0.15577873E-01	0.47527873E-04	0.25404049E-04	0.15392447E-04
0.21447530E-01	0.92790148E-02	-0.94694444E-02	0.29060942E-04	0.15392470E-04	0.95940851E-05

THERE WAS NO RECTIFICATION AT THIS TIME

```
TIME FROM JD 2433282.5 =   0.55950000E 04    DAYS          ,     0.56465969E 05   SEC (U.T.)
R VECTOR (1950)        =   0.57196070E 04   0.31759088E 04     0.33429365E 04
V VECTOR (1950)        =  -0.34946938E 01  -0.52635898E 00     0.65629485E 01
R VECTOR (DATE)        =   0.57040812E 04   0.31949766E 04     0.33512842E 04
V VECTOR (DATE)        =  -0.35024555E 01  -0.53814517E 00     0.65578529E 01

  FLOYC  OBSERVES
    RELATIVE POSITION  =   0.12130013E 04   0.19624383E 04    -0.99234814E 03
    RELATIVE VELOCITY  =  -0.34048156E 01  -0.85385368E 00     0.65629485E 01
    R, RDOT, AZ, ELEV  =   0.25114321E 04  -0.49049349E 01     0.14010343E 03   0.12699329E 02

TIME FROM JD 2433282.5 =   0.55950000E 04    DAYS          ,     0.56475960E 05   SEC (U.T.)
R VECTOR (1950)        =   0.56844064E 04   0.31704910E 04     0.34083356E 04
V VECTOR (1950)        =  -0.35519519E 01  -0.55822408E 00     0.65289693E 01
R VECTOR (DATE)        =   0.56688038E 04   0.31894401E 04     0.34166320E 04
V VECTOR (DATE)        =  -0.35595573E 01  -0.57020115E 00     0.65237901E 01

  FLOYC  OBSERVES
    RELATIVE POSITION  =   0.11786219E 04   0.19536344E 04    -0.92700034E 03
    RELATIVE VELOCITY  =  -0.34618356E 01  -0.88565331E 00     0.65289693E 01
    R, RDOT, AZ, ELEV  =   0.24627559E 04  -0.48168780E 01     0.13902282E 03   0.13425324E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0     ) =    0.55950000E 04    0.73065695E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =    0.56475960E 05    0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )   =    0.54659796E 04    0.31009080E 04    0.35025183E 04
VELOCITY VECTOR ( DATE )   =   -0.38297505E 01   -0.71397462E 00    0.64313924E 01
DELTA RADIUS               =   -0.20282417E 03   -0.88532024E 02    0.85886231E 02
DELTA VELOCITY             =   -0.27019325E 00   -0.14377347E 00   -0.92397689E-01
```

THE COVARIANCE MATRIX FOR THE ERRORS IN STATE ** P **

0.50503440E 02	0.21668605E 02	-0.21314460E 02	0.66992119E-01	0.35732130E-01	0.22820565E-01
0.21668588E 02	0.96341556E 01	-0.92659634E 01	0.29005693E-01	0.15285064E-01	0.99287765E-02
-0.21314445E 02	-0.92659714E 01	0.90789179E 01	-0.28341262E-01	-0.15024379E-01	-0.96074095E-02
0.66992157E-01	0.29005725E-01	-0.28341301E-01	0.89114470E-04	0.47343215E-04	0.30452314E-04
0.35732103E-01	0.15285066E-01	-0.15024379E-01	0.47343187E-04	0.25417169E-04	0.16194629E-04
0.22820577E-01	0.99287916E-02	-0.96074303E-02	0.30452311E-04	0.16194652E-04	0.10594193E-04

THERE WAS NO RECTIFICATION AT THIS TIME

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS        ,     0.56485971E 05   SEC (U.T.)
R VECTOR (1950)        =   0.56485610E 04   0.31647428E 04    0.34735259E 04
V VECTOR (1950)        =  -0.36089590E 01  -0.59039212E 00    0.64942687E 01
R VECTOR (DATE)        =   0.56328831E 04   0.31835710E 04    0.34817700E 04
V VECTOR (DATE)        =  -0.36164069E 01  -0.60225921E 00    0.64890064E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.11436054E 04   0.19444845E 04   -0.86186237E 03
    RELATIVE VELOCITY  =  -0.35186034E 01  -0.91745540E 00    0.64942687E 01
    R, RDOT, AZ, ELEV  =   0.24148830E 04  -0.47228115E 01    0.13788906E 03   0.14162094E 02


TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS        ,     0.56495983E 05   SEC (U.T.)
R VECTOR (1950)        =   0.56121469E 04   0.31586760E 04    0.35383656E 04
V VECTOR (1950)        =  -0.36655900E 01  -0.62189363E 00    0.64589205E 01
R VECTOR (DATE)        =   0.55963951E 04   0.31773814E 04    0.35465566E 04
V VECTOR (DATE)        =  -0.36728797E 01  -0.63424949E 00    0.64535755E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.11080228E 04   0.19350195E 04   -0.79707578E 03
    RELATIVE VELOCITY  =  -0.35749955E 01  -0.94919090E 00    0.64589205E 01
    R, RDOT, AZ, ELEV  =   0.23679832E 04  -0.46225531E 01    0.13670107E 03   0.14907388E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0     ) =     0.55950000E 04    0.73088869E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =       0.56495983E 05    0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )    =     0.53893516E 04    0.30863763E 04    0.36300257E 04
VELOCITY VECTOR ( DATE )    =    -0.39446667E 01   -0.77950037E 00    0.63554542E 01
DELTA RADIUS                =    -0.20704347E 03   -0.91005183E 02    0.83469116E 02
DELTA VELOCITY              =    -0.27178703E 00   -0.14525088E 00   -0.98121273E-01
```

THE COVARIANCE MATRIX FOR THE ERRORS IN STATE ** P **

0.52071372E 02	0.22489069E 02	-0.20934083E 02	0.68046989E-01	0.36451304E-01	0.24456204E-01
0.22489050E 02	0.10043392E 02	-0.91555604E 01	0.29645170E-01	0.15692144E-01	0.10701606E-01
-0.20934067E 02	-0.91555680E 01	0.84994175E 01	-0.27417826E-01	-0.14594989E-01	-0.98025844E-02
0.68047026E-01	0.29645203E-01	-0.27417866E-01	0.89168670E-04	0.47574818E-04	0.32140299E-04
0.36451277E-01	0.15692146E-01	-0.14594991E-01	0.47574790E-04	0.25656687E-04	0.17164587E-04
0.24456215E-01	0.10701622E-01	-0.98026056E-02	0.32140296E-04	0.17164610E-04	0.11768177E-04

THERE WAS NO RECTIFICATION AT TH·S TIME

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS           ,   0.56505973E 05   SEC (U.T.)
R VECTOR (1950)        =   0.55752445E 04   0.31523047E 04   0.36027157E 04
V VECTOR (1950)        =  -0.37217233E 01  -0.65355989E 00   0.64230031E 01
R VECTOR (DATE)        =   0.55594208E 04   0.31708857E 04   0.36108530E 04
V VECTOR (DATE)        =  -0.37288544E 01  -0.66610283E 00   0.64175761E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.10719550E 04   0.19252536E 04  -0.73277942E 03
    RELATIVE VELOCITY  =  -0.36308904E 01  -0.98079105E 00   0.64230031E 01
    R, RDOT, AZ, ELEV  =   0.23222090E 04  -0.45159876E 01   0.13545869E 03   0.15658882E 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS           ,   0.56515964E 05   SEC (U.T.)
R VECTOR (1950)        =   0.55377831E 04   0.31456172E 04   0.36667040E 04
V VECTOR (1950)        =  -0.37774709E 01  -0.68515338E 00   0.63864489E 01
R VECTOR (DATE)        =   0.55218889E 04   0.31640721E 04   0.36747866E 04
V VECTOR (DATE)        =  -0.37844428E 01  -0.69788211E 00   0.63809406E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.10353318E 04   0.19151704E 04  -0.66884579E 03
    RELATIVE VELOCITY  =  -0.36863996E 01  -0.10123183E 01   0.63864489E 01
    R, RDOT, AZ, ELEV  =   0.22775303E 04  -0.44025574E 01   0.13415634E 03   0.16417032E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0    ) =    0.55950000E 04   0.73111996E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =    0.56515964E 05   0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )  =    0.53157545E 04   0.30727964E 04   0.37537574E 04
VELOCITY VECTOR ( DATE )  =   -0.40511351E 01  -0.84105811E 00   0.62794219E 01
DELTA RADIUS              =   -0.20613446E 03  -0.91275685E 02   0.78970818E 02
DELTA VELOCITY            =   -0.26669234E 00  -0.14317600E 00  -0.10151868E 00


THE COVARIANCE MATRIX FOR THE ERRORS IN STATE  ** P **


 0.53543014E 02   0.23280698E 02  -0.20470988E 02   0.68938001E-01   0.37094244E-01   0.26104194E-01
 0.23280679E 02   0.10446481E 02  -0.90087588E 01   0.30224794E-01   0.16073346E-01   0.11490874E-01
-0.20470972E 02  -0.90087661E 01   0.79103405E 01  -0.26411400E-01  -0.14119589E-01  -0.99452741E-02
 0.68938036E-01   0.30224826E-01  -0.26411438E-01   0.88998826E-04   0.47695421E-04   0.33790500E-04
 0.37094218E-01   0.16073348E-01  -0.14119591E-01   0.47695395E-04   0.25842598E-04   0.18125467E-04
 0.26104205E-01   0.11490891E-01  -0.99452949E-02   0.33790495E-04   0.18125490E-04   0.13008072E-04

THERE WAS NO RECTIFICATION AT THIS TIME
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS        ,   0.56525976E 05   SEC (U.T.)
R VECTOR (1950)        =   0.54996875E 04   0.31335998E 04   0.37304546E 04
V VECTOR (1950)        =  -0.38329403E 01  -0.71673552E 00   0.63491848E 01
R VECTOR (DATE)        =   0.54837243E 04   0.31569262E 04   0.37384817E 04
V VECTOR (DATE)        =  -0.38397519E 01  -0.72964909E 00   0.63435956E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.99807977E 03   0.19047511E 04  -0.60515070E 03
    RELATIVE VELOCITY  =  -0.37416302E 01  -0.10438339E 01   0.63491848E 01
    R, RDOT, AZ, ELEV  =   0.22339309E 04  -0.42816465E 01   0.13278796E 03   0.17182111E 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS        ,   0.56535987E 05   SEC (U.T.)
R VECTOR (1950)        =   0.54610389E 04   0.31312666E 04   0.37938289E 04
V VECTOR (1950)        =  -0.38880100E 01  -0.74823752E 00   0.63112900E 01
R VECTOR (DATE)        =   0.54450083E 04   0.31494629E 04   0.38017996E 04
V VECTOR (DATE)        =  -0.38946608E 01  -0.76133459E 00   0.63056204E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.96027856E 03   0.18940150E 04  -0.54183276E 03
    RELATIVE VELOCITY  =  -0.37964613E 01  -0.10752692E 01   0.63112900E 01
    R, RDOT, AZ, ELEV  =   0.21915771E 04  -0.41531278E 01   0.13135302E 03   0.17951028E 02

UNDRFLOW AT 23550 IN MQ
```

```
EPOCH (REL. TO 1950.0    ) =    0.55950000E 04   0.73135170E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =    0.56535987E 05   0.55950000E 04 (DAYS  U.T.)

POSITICN VECTOR ( DATE )  =    0.52488876E 04   0.30619112E 04   0.38730161E 04
VELOCITY VECTOR ( DATE )  =   -0.41447970E 01  -0.89626732E 00   0.62052931E 01
DELTA RADIUS              =   -0.19612067E 03  -0.87551635E 02   0.71216523E 02
DELTA VELOCITY            =   -0.25013625E 00  -0.13493274E 00  -0.10032725E 00


THE COVARIANCE MATRIX FOR THE ERRORS IN STATE  ** P **


 0.53965287E 02   0.23625135E 02  -0.19577016E 02   0.68457765E-01   0.37010978E-01   0.27285503E-0
 0.23625118E 02   0.10659925E 02  -0.86720572E 01   0.30213969E-01   0.16143176E-01   0.12087073E-0
-0.19577000E 02  -0.86720637E 01   0.71863697E 01  -0.24882422E-01  -0.13360975E-01  -0.98577116E-0
 0.68457795E-01   0.30213997E-01  -0.24882457E-01   0.87076572E-04   0.46861740E-04   0.34793643E-0
 0.37010953E-01   0.16143177E-01  -0.13360977E-01   0.46881719E-04   0.25531579E-04   0.18749964E-0
 0.27285512E-01   0.12087089E-01  -0.98577316E-02   0.34793636E-04   0.18749985E-04   0.14076574E-0

THERE WAS NO RECTIFICATION AT THIS TIME
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS        ,   0.56545978E 05   SEC (U.T.)
R VECTOR (1950)        =   0.54219222E 04   0.31236345E 04   0.38566916E 04
V VECTOR (1950)        =  -0.39425623E 01  -0.77959149E 00   0.62728487E 01
R VECTOR (DATE)        =   0.54058260E 04   0.31416990E 04   0.38646052E 04
V VECTOR (DATE)        =  -0.39490518E 01  -0.79287032E 00   0.62670994E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.92201239E 03   0.18829836E 04  -0.47902716E 03
    RELATIVE VELOCITY  =  -0.38507753E 01  -0.11065563E 01   0.62728487E 01
    R, RDOT, AZ, ELEV  =   0.21506281E 04  -0.40169450E 01   0.12985155E 03   0.18720369E 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS        ,   0.56555968E 05   SEC (U.T.)
R VECTOR (1950)        =   0.53822625E 04   0.31156895E 04   0.39191671E 04
V VECTOR (1950)        =  -0.39967054E 01  -0.81085913E 00   0.62337879E 01
R VECTOR (DATE)        =   0.53661022E 04   0.31336205E 04   0.39270229E 04
V VECTOR (DATE)        =  -0.40030332E 01  -0.82431834E 00   0.62279596E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.88320697E 03   0.18716383E 04  -0.41660950E 03
    RELATIVE VELOCITY  =  -0.39046801E 01  -0.11377570E 01   0.62337879E 01
    R, RDOT, AZ, ELEV  =   0.21110774E 04  -0.38725086E 01   0.12827748E 03   0.19489499E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0     ) =    0.55950000E 04   0.73158297E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =    0.56555968E 05   0.55950000E 04 (DAYS  U.T.)

POSITICN VECTOR ( DATE )  =    0.51931786E 04   0.30556772E 04   0.39863504E 04
VELOCITY VECTOR ( DATE )  =   -0.42205738E 01  -0.94227163E 00   0.61360451E 01
DELTA RADIUS              -   -0.17292360E 03  -0.77943280E 02   0.59327497E 02
DELTA VELOCITY            =   -0.21754062E 00  -0.11795229E 00  -0.91914522E-01


THE COVARIANCE MATRIX FOR THE ERRORS IN STATE  ** P **


 0.52066952E 02   0.22949787E 02  -0.17875780E 02   0.65079889E-01   0.35364902E-01   0.27277407E-0
 0.22949772E 02   0.10426138E 02  -0.79745257E 01   0.28923870E-01   0.15524390E-01   0.12166162E-0
-0.17875765E 02  -0.79745315E 01   0.62225703E 01  -0.22384993E-01  -0.12073358E-01  -0.93187962E-0
 0.65079913E-01   0.28923890E-01  -0.22385023E-01   0.81574717E-04   0.44131017E-04   0.34275307E-0
 0.35364880E-01   0.15524390E-01  -0.12073361E-01   0.44131005E-04   0.24173563E-04   0.18560316E-0
 0.27277412E-01   0.12166175E-01  -0.93188144E-02   0.34275300E-04   0.18560334E-04   0.14570445E-0


THERE WAS NO RECTIFICATION AT THIS TIME
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS         ,    0.56565979E 05   SEC (U.T.)
R VECTOR (1950)        =   0.53419804E 04   0.31074153E 04    0.39813768E 04
V VECTOR (1950)        =  -0.40505440E 01  -0.84210125E 00    0.61940300E 01
R VECTOR (DATE)        =   0.53257576E 04   0.31252107E 04    0.39891739E 04
V VECTOR (DATE)        =  -0.40567091E 01  -0.85573982E 00    0.61881232E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.84378571E 03   0.18599529E 04   -0.35445853E 03
    RELATIVE VELOCITY  =  -0.39582799E 01  -0.11689318E 01    0.61940300E 01
    R, RDOT, AZ, ELEV  =   0.20729302E 04  -0.37191916E 01    0.12662441E 03   0.20257490F 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS         ,    0.56575991E 05   SEC (U.T.)
R VECTOR (1950)        =   0.53011615E 04   0.30988288E 04    0.40431855E 04
V VECTOR (1950)        =  -0.41039604E 01  -0.87324995E 00    0.61536592E 01
R VECTOR (DATE)        =   0.52848778E 04   0.31164867E 04    0.40509230E 04
V VECTOR (DATE)        =  -0.41099622E 01  -0.88706645E 00    0.61476744E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.80383026E 03   0.18479588E 04   -0.29270938E 03
    RELATIVE VELOCITY  =  -0.40114579E 01  -0.12000131E 01    0.61536592E 01
    R, RDOT, AZ, ELEV  =   0.20363629E 04  -0.35569986E 01    0.12489245E 03   0.21019698E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0     ) =     0.55950000E 04    0.73181471E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =     C.56575991E 05    0.55950000E 04 (DAYS  U.T.)

POSITICN VECTOR ( DATE )  =     0.51482107E 04    0.30541107E 04    0.40949998E 04
VELOCITY VECTOR ( DATE )  =    -0.42797135E 01   -0.97963039E 00    0.60720487E 01
DELTA RADIUS              =    -0.13666706E 03   -0.62376046E 02    0.44076783E 02
DELTA VELOCITY            =    -C.16975137E 00   -0.92563935E-01   -0.75625651E-01
```

THE COVARIANCE MATRIX FOR THE ERRORS IN STATE ** P **

0.47004577E 02	0.20856617E 02	-0.15218746E 02	0.57887391F-01	0.31634774E-01	0.25500763E-0
0.20856608E 02	0.95580225E 01	-0.68419878E 01	0.25917916E-01	0.13969675E-01	0.11458935E-0
-0.15218735E 02	-0.68419928E 01	0.50140782E 01	-0.18777558E-01	-0.10171735E-01	-0.82048214E-0
0.57887404E-01	0.25917924E-01	-0.18777580E-01	0.71515002E-04	0.38882907E-04	0.31585479E-0
0.31634757E-01	0.13969673E-01	-0.10171738E-01	0.38882907E-04	0.21448585E-04	0.17191626E-0
0.25500765E-01	0.11458945E-01	-0.82048371E-02	0.31585471E-04	0.17191639E-04	0.14114013E-0

THERE WAS NO RECTIFICATION AT THIS TIME

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS           ,   0.56585981E 05   SEC (U.T.)
R VECTOR (1950)        =   0.52598957E 04   0.30899496E 04   0.41044609E 04
V VECTOR (1950)        =  -0.41568404E 01  -0.90423810E 00   0.61127656E 01
R VECTOR (DATE)        =   0.52435529E 04   0.31074686E 04   0.41121383E 04
V VECTOR (DATE)        =  -0.41626787E 01  -0.91823074E 00   0.61067036E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.76343097E 03   0.18356759E 04  -0.23149408E 03
    RELATIVE VELOCITY  =  -0.40640998E 01  -0.12309337E 01   0.61127656E 01
    R, RDOT, AZ, ELEV  =   0.20015300E 04  -0.33860717E 01   0.12308309E 03   0.21771341E 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS           ,   0.56595972E 05   SEC (U.T.)
R VECTOR (1950)        =   0.52181035E 04   0.30807613E 04   0.41653250E 04
V VECTOR (1950)        =  -0.42092896E 01  -0.93512681E 00   0.60712712E 01
R VECTOR (DATE)        =   0.52017032E 04   0.30981396E 04   0.41729413E 04
V VECTOR (DATE)        =  -0.42149638E 01  -0.94929413E 00   0.60651327E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.72250916E 03   0.18230827E 04  -0.17069104E 03
    RELATIVE VELOCITY  =  -0.41163109E 01  -0.12617548E 01   0.60712712E 01
    R, RDOT, AZ, ELEV  =   0.19684474E 04  -0.32059110E 01   0.12119064E 03   0.22510196E 02

UNDRFLOW AT 33550 IN MQ
```

EPOCH (REL. TO 1950.0)	=	0.55950000E 04	0.73204598E 00	(DAYS,DAYS)
EPOCH (REL. JD 2433282.5)	=	0.56595972E 05	0.55950000E 04	(DAYS U.T.)

POSITION VECTOR (DATE)	=	0.51072400E 04	0.30542355E 04	0.42013411E 04
VELOCITY VECTOR (DATE)	=	-0.43310577E 01	-0.10130141E 01	0.60104529E 01
DELTA RADIUS	=	-0.94463189E 02	-0.43904063E 02	0.28399769E 02
DELTA VELOCITY	=	-0.11609398E 00	-0.63719989E-01	-0.54679736E-01

THE COVARIANCE MATRIX FOR THE ERRORS IN STATE ** P **

0.39310274E 02	0.17551609E 02	-0.11951990E 02	0.47697693E-01	0.26233403E-01	0.22071700E-01
0.17551605E 02	0.81333916E 01	-0.54196041E 01	0.21523152E-01	0.11642594E-01	0.99994075E-02
-0.11951981E 02	-0.54196083E 01	0.37221015E 01	-0.14531161E-01	-0.79031227E-02	-0.66537805E-02
0.47697695E-01	0.21523148E-01	-0.14531175E-01	0.58095639E-04	0.31751194E-04	0.26951801E-04
0.26233391E-01	0.11642591E-01	-0.79031255E-02	0.31751207E-04	0.17669641E-04	0.14754977E-04
0.22071698E-01	0.99994123E-02	-0.66537931E-02	0.26961794E-04	0.14754986E-04	0.12671512E-04

THERE WAS NO RECTIFICATION AT THIS TIME

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS         ,     [illegible]505983E 05   SEC (U.T.)
R VECTOR (1950)        =   0.51757020E 04   0.30712449E 04   0.42258956E 04
V VECTOR (1950)        =  -0.42614084E 01  -0.96597592E 00   0.60290934E 01
R VECTOR (DATE)        =   0.51592457E 04   0.30884805E 04   0.42334501E 04
V VECTOR (DATE)        =  -0.42669177E 01  -0.98031680E 00   0.60228788E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.68C98322E 03   0.18101556E 04  -0.11018225E 03
    RELATIVE VELOCITY  =  -0.41681914E 01  -0.12925360E 01   0.60290934E 01
    R, RDOT, AZ, ELEV  =   0.19371478E C4  -0.30160108E 01   0.11920928E 03   0.23233528E 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS         ,     0.56615995E 05   SEC (U.T.)
R VECTOR (1950)        =   0.51327806E 04   0.30614203E 04   0.42860410E 04
V VECTOR (1950)        =  -0.43130836E C1  -0.99671865E 00   0.59863222E 01
R VECTOR (DATE)        =   0.51162700E 04   0.30785114E 04   0.42935330E 04
V VECTOR (CATE)        =  -0.43184276E 01  -0.10112316E 01   0.59800321E 01

  FLOYD  OBSERVES
    RFLATIVE POSITION  =   0.63894153E 03   0.17969191E 04  -0.50099365E 02
    RELATIVE VELOCITY  =  -0.42196285E 01  -0.13232106E 01   0.59863222E 01
    R, RDOT, AZ, ELEV  =   0.19077931E 04  -0.28167148E 01   0.11714191E 03   0.23935227E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0     ) =    0.55950000E 04    0.73227772E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =     0.56615995E 05    0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )  =    0.50608483E 04    0.30519287E 04    0.43087778E 04
VELOCITY VECTOR ( DATE )  =   -0.43861717E 01   -0.10487272E 01    0.59460384E 01
DELTA RADIUS              =   -0.55421723E 02   -0.26582695E 02    0.15244760E 02
DELTA VELOCITY            =   -0.67744167E-01   -0.37495593E-01   -0.33993757E-01


THE COVARIANCE MATRIX FOR THE ERRORS IN STATE  ** P **


 0.30775307E 02   0.13817250E 02  -0.87369738E 01   0.36785079E-01   0.20382203E-01   0.17876385E-01
 0.13817251E 02   0.64934383E 01  -0.39998329E 01   0.16736404E-01   0.90778975E-02   0.81712750E-02
-0.87369676E 01  -0.39998367E 01   0.25702932E 01  -0.10465969E-01  -0.57114771E-02  -0.50127829E-02
 0.36785074E-01   0.16736389E-01  -0.10465976E-01   0.44185464E-04   0.24279909E-04   0.21548346E-04
 0.20382196E-01   0.90778922E-02  -0.57114793E-02   0.24279931E-04   0.13667163E-04   0.11861419E-04
 0.17876380E-01   0.81712757E-02  -0.50127931E-02   0.21548340E-04   0.11861424E-04   0.10662100E-04


THE TRAJECTORY HAS BEEN RECTIFIED

POSITION VECTOR (R DATE)  =    0.50608483E 04    0.30519287E 04    0.43087778E 04
VELOCITY VECTOR (V DATE)  =   -0.43861717E 01   -0.10487272E 01    0.59460384E 01


ELLIPTIC ORBIT ELEMENTS

SEMIMAJOR AXIS        (KM)  =    0.74776414E 04
ECCENTRICITY                =    0.22280758E-01
TRUE ANOMOLY DATE     (DEG) =    0.10735342E 02
R ASC OF ASC NODE     (DEG) =    0.24827538E 02
ARG OF PERIAPSE       (DEG) =    0.25826448E 02
ORBIT INCLINATION     (DEG) =    0.81488110E 02
DEL NODE PER REV      (RAD) =   -0.10995342E-02
DEL APSE PER REV      (RAD) =   -0.33074063E-02
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS      ,     0.56625985E 05   SEC (U.T.)
R VECTOR (1950)        =   0.50325674E 04   0.30239071E 04   0.43634204E 04
V VECTOR (1950)        =  -0.44343504E 01  -0.10657655E 01   0.59071504E 01
R VECTOR (DATE)        =   0.50160704E 04   0.30406620E 04   0.43707662E 04
V VECTOR (DATE)        =  -0.44393473E 01  -0.10806826E 01   0.59006833E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.53967694E 03   0.17558076E 04   0.27133789E 02
    RELATIVE VELOCITY  =  -0.43406573E 01  -0.13921892E 01   0.59071504E 01
    R, RDOT, AZ, ELEV  =   0.18370758E 04  -0.25185055E 01   0.11343114E 03   0.24090819E 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS      ,     0.56635976E 05   SEC (U.T.)
R VECTOR (1950)        =   0.49880104E 04   0.30131059E 04   0.44222139E 04
V VECTOR (1950)        =  -0.44852745E 01  -0.10964456E 01   0.58623946E 01
R VECTOR (DATE)        =   0.49714643E 04   0.30297110E 04   0.44294946E 04
V VECTOR (DATE)        =  -0.44901034E 01  -0.11115323E 01   0.58558532E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.49600824E 03   0.17415951E 04   0.85862244E 02
    RELATIVE VELOCITY  =  -0.43913436E 01  -0.14228010E 01   0.58623946E 01
    R, RDOT, AZ, ELEV  =   0.18128844E 04  -0.22906741E 01   0.11111772E 03   0.24695814E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0     ) =    0.55950000E 04   0.73250899E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =    0.56635976E 05   0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )  =    0.50080609E 04   0.30463147E 04   0.44195542E 04
VELOCITY VECTOR ( DATE )  =   -0.44470568E 01  -0.10877775E 01   0.58775209E 01
DELTA RADIUS              =    0.36596663E 02   0.16603726E 02  -0.99403938E 01
DELTA VELOCITY            =    0.43046511E-01   0.23754710E-01   0.21667763E-01


[illegible] COVARIANCE MATRIX FOR THE ERRORS IN STATE  ** P **


 0.[illegible]E 02   0.97573962E 01  -0.56908583E 01   0.25478090E-01   0.14252418E-01   0.13020126E-01
 0.[illegible]E 01   0.46806981E 01  -0.26361292E 01   0.11698631E-01   0.63482089E-02   0.60142842E-02
-0.[illegible]E 01  -0.26361330E 01   0.15877198E 01  -0.67141122E-02  -0.36698815E-02  -0.33551637E-02
 0.25[illegible]E-01   0.11698609E-01  -0.67141157E-02   0.30198223E-04   0.16688244E-04   0.15504842E-04
 0.14252414E-01   0.63482022E-02  -0.36698832E-02   0.16688273E-04   0.95566515E-05   0.85903778E-05
 0.13[illegible]E-01   0.60142817E-02  -0.33551721E-02   0.15504837E-04   0.85903790E-05   0.81086514E-05


THE TRAJECTORY HAS BEEN RECTIFIED

POSITION VECTOR (R DATE)  =    0.50080609E 04   0.30463147E 04   0.44195542E 04
VELOCITY VECTOR (V DATE)  =   -0.44470568E 01  -0.10877775E 01   0.58775209E 01


[illegible] ORBIT ELEMENTS

SEMIMAJOR AXIS        (KM)  =    0.75082830E 04
ECCENTRICITY                =    0.23373676E-01
TRUE ANOMOLY DATE     (DEG) =    0.18223415E 02
R ASC OF ASC NODE     (DEG) =    0.24828821E 02
ARG OF PERIAPSE       (DEG) =    0.19274548E 02
ORBIT INCLINATION     (DEG) =    0.81483543E 02
DEL NODE PER REV      (RAD) =   -0.10912677E-02
DEL APSE PER REV      (RAD) =   -0.32903638E-02
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS           ,   0.56645987E 05   SEC (U.T.)
R VECTOR (1950)        =   0.49791888E 04   0.30180441E 04   0.44736817E 04
V VECTOR (1950)        =  -0.44943137E 01  -0.11041815E 01   0.58378555E 01
R VECTOR (DATE)        =   0.49625513E 04   0.30346189E 04   0.44809496E 04
V VECTOR (DATE)        =  -0.44990809E 01  -0.11192980E 01   0.58313009E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.48803638E 03   0.17432377E 04   0.13731726E 03
    RELATIVE VELOCITY  =  -0.44001446E 01  -0.14304683E 01   0.58378555E 01
    R, RDOT, AZ, ELEV  =   0.18154650E 04  -0.21148513E 01   0.10971661E 03   0.25732176E 02


TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS           ,   0.56655999E 05   SEC (U.T.)
R VECTOR (1950)        =   0.49339447E 04   0.30068379E 04   0.45318992E 04
V VECTOR (1950)        =  -0.45442219E 01  -0.11345145E 01   0.57924080E 01
R VECTOR (DATE)        =   0.49172604E 04   0.30232605E 04   0.45391011E 04
V VECTOR (DATE)        =  -0.45488213E 01  -0.11497971E 01   0.57857806E 01

  FLUYD  OBSERVES
    RELATIVE POSITION  =   0.44369025E 03   0.17286098E 04   0.19546875E 03
    RELATIVE VELOCITY  =  -0.44498145E 01  -0.14607323E 01   0.57924080E 01
    R, RDOT. AZ, ELEV  =   0.17953164F 04  -0.18755139E 01   0.10728010E 03   0.26277400E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0    ) =   0.55950000E 04   0.73274073E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =   0.56655999E 05   0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )  =   0.49418344E 04   0.30345012E 04   0.45328723E 04
VELOCITY VECTOR ( DATE )  =  -0.45203459E 01  -0.11340105E 01   0.58007791E 01
DELTA RADIUS              =   0.24573993E 02   0.11240703E 02  -0.62298212E 01
DELTA VELOCITY            =   0.28475297E-01   0.15786513E-01   0.14998540E-01


THE COVARIANCE MATRIX FOR THE ERRORS IN STATE  ** P **


 0.15982680E 02   0.72239416E 01  -0.38575959E 01   0.18516736E-01   0.10469589E-01   0.99383311E-02
 0.72239501E 01   0.35409497E 01  -0.18087653E 01   0.85789828E-02   0.46546090E-02   0.46383984E-02
-0.38575934E 01  -0.18087692E 01   0.10248382E 01  -0.44796237E-02  -0.24487822E-02  -0.23260446E-02
 0.18516726E-01   0.85789567E-02  -0.44796253E-02   0.21662593E-04   0.12041932E-04   0.11696302E-04
 0.10469588E-01   0.46546016E-02  -0.24487838E-02   0.12041964E-04   0.70372117E-05   0.65254550E-05
 0.99383228E-02   0.46383940E-02  -0.23260520E-02   0.11696297E-04   0.65254546E-05   0.64575960E-05

THE TRAJECTORY HAS BEEN RECTIFIED

POSITION VECTOR (R DATE)  =   0.49418344E 04   0.30345012E 04   0.45328723E 04
VELOCITY VECTOR (V DATE)  =  -0.45203459E 01  -0.11340105E 01   0.58007791E 01

ELLIPTIC ORBIT ELEMENTS

SEMIMAJOR AXIS       (KM)  =   0.75293311E 04
ECCENTRICITY               =   0.24309775E-01
TRUE ANOMOLY DATE    (DEG) =   0.23260010E 02
R ASC OF ASC NODE    (DEG) =   0.24828077E 02
ARC OF PERIAPSE      (DEG) =   0.15254175E 02
ORBIT INCLINATION    (DEG) =   0.81481152E 02
DEL NODE PER REV     (RAD) =  -0.10855744E-02
DEL APSE PER REV     (RAD) =  -0.32621163E-02
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,   0.56665989E 05   SEC (U.T.)
R VECTOR (1950)        =   0.49124175E 04   0.30060023E 04   0.45861779E 04
V VECTOR (1950)        =  -0.45665521E 01  -0.11496529E 01   0.57603918E 01
R VECTOR (DATE)        =   0.48956572E 04   0.30223521E 04   0.45933485E 04
V VECTOR (DATE)        =  -0.45710540E 01  -0.11650097E 01   0.57537317E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.42303180E 03   0.17244421E 04   0.24971613E 03
    RELATIVE VELOCITY  =  -0.44719070E 01  -0.14758019E 01   0.57603918E 01
    R, RDOT, AZ, ELEV  =   0.17930462E 04  -0.16721438E 01   0.10544583E 03   0.27123124E 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,   0.56675980E 05   SEC (U.T.)
R VECTOR (1950)        =   0.48665507E 04   0.29943670E 04   0.46434984E 04
V VECTOR (1950)        =  -0.46152887E 01  -0.11795580E 01   0.57142836E 01
R VECTOR (DATE)        =   0.48497462E 04   0.30105626E 04   0.46506021E 04
V VECTOR (DATE)        =  -0.46196232E 01  -0.11950769E 01   0.57075524E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.37806763E 03   0.17093941E 04   0.30696967E 03
    RELATIVE VELOCITY  =  -0.45204060E 01  -0.15056379E 01   0.57142836E 01
    R, RDOT, AZ, ELEV  =   0.17774121E 04  -0.14226504E 01   0.10289645E 03   0.27580292E 02

UNDRFLOW AT 33550 IN MQ
```

EPOCH (REL. TO 1950.0) =	0.55950000E 04	0.73297200E 00	(DAYS,DAYS)
EPOCH (REL. JD 2433282.5) =	0.56675980E 05	0.55950000E 04	(DAYS U.T.)

POSITION VECTOR (DATE) =	0.48670094E 04	0.30185294E 04	0.46465382E 04
VELOCITY VECTOR (DATE) =	-0.45999113E 01	-0.11841008E 01	0.57184050E 01
DELTA RADIUS =	0.17263239E 02	0.79669062E 01	-0.40638185E 01
DELTA VELOCITY =	0.19711917E-01	0.10976126E-01	0.10852667E-01

THE COVARIANCE MATRIX FOR THE ERRORS IN STATE ** P **

0.12160710E 02	0.55047125E 01	-0.26676063E 01	0.13865820E-01	0.79352189E-02	0.78077641E-02
0.55047222E 01	0.27604813E 01	-0.12667426E 01	0.64806812E-02	0.35130504E-02	0.36810580E-02
-0.26676043E 01	-0.12667466E 01	0.68110189E 00	-0.30458396E-02	-0.16613025E-02	-0.16348530E-02
0.13865809E-01	0.64806534E-02	-0.30458405E-02	0.16016854E-04	0.89581793E-05	0.90854431E-05
0.79352186E-02	0.35130425E-02	-0.16613040E-02	0.89582120E-05	0.53624179E-05	0.51068961E-05
0.78077554E-02	0.36810527E-02	-0.16348601E-02	0.90854387E-05	0.51068949E-05	0.52905671E-05

THE TRAJECTORY HAS BEEN RECTIFIED

POSITION VECTOR (R DATE) =	0.48670094E 04	0.30185294E 04	0.46465382E 04
VELOCITY VECTOR (V DATE) =	-0.45999113E 01	-0.11841008E 01	0.57184050E 01

ELLIPTIC ORBIT ELEMENTS

SEMIMAJOR AXIS	(KM) =	0.75442856E 04
ECCENTRICITY	=	0.25043313E-01
TRUE ANOMOLY DATE	(DEG) =	0.26959002E 02
R ASC OF ASC NODE	(DEG) =	0.24826047E 02
ARG OF PERIAPSE	(DEG) =	0.12615098E 02
ORBIT INCLINATION	(DEG) =	0.81479958E 02
DEL NODE PER REV	(RAD) =	-0.10815037E-02
DEL APSE PER REV	(RAD) =	-0.32493206E-02

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS            ,    0.56685971E 05   SEC (U.T.)
R VECTOR (1950)        =   0.48368821E 04   0.29897721E 04       0.46990955E 04
V VECTOR (1950)        =  -0.46452088E 01  -0.11990403E 01       0.56771411E 01
R VECTOR (DATE)        =   0.48200123E 04   0.30058676E 04       0.47061559E 04
V VECTOR (DATE)        =  -0.46494240E 01  -0.12146587E 01       0.56703662E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.34928156E 03   0.17014458E 04       0.36252350E 03
    RELATIVE VELOCITY  =  -0.45500889E 01  -0.15250512E 01       0.56771411E 01
    R, RDOT, AZ, ELEV  =   0.17743559E 04  -0.11981570E 01       0.10074713E 03   0.28255942E 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS            ,    0.56695961E 05   SEC (U.T.)
R VECTOR (1950)        =   0.47902346E 04   0.29776451E 04       0.47555800E 04
V VECTOR (1950)        =  -0.46928989E 01  -0.12286015E 01       0.56301946E 01
R VECTOR (DATE)        =   0.47733235E 04   0.29935837E 04       0.47625723E 04
V VECTOR (DATE)        =  -0.46969465E 01  -0.12443786E 01       0.56233501E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.30354456E 03   0.16859047E 04       0.41893994E 03
    RELATIVE VELOCITY  =  -0.45975414E 01  -0.15545429E 01       0.56301946E 01
    R, RDOT, AZ, ELEV  =   0.17634978E 04  -0.93998215E 00       0.98096076E 02   0.28602905E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0    ) =    0.55950000E 04   0.73320326E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =     0.56695961E 05   0.55950000E 04 (DAYS  U.T.)

POSITICN VECTOR ( DATE )  =     0.47887496E 04   0.30007676E 04   0.47592227E 04
VELOCITY VECTOR ( DATE )  =    -0.46795872E 01  -0.12346685E 01   0.56333339E 01
DELTA RADIUS              =     0.15426056E 02   0.71839305E 01  -0.33496431E 01
DELTA VELOCITY            =     0.17359211E-01   0.97100786E-02   0.99838180E-02

THE COVARIANCE MATRIX FOR THE ERRORS IN STATE  ** P **

 0.94938983E 01   0.42978318E 01  -0.18688515E 01   0.10649736E-01   0.61782866E-02   0.62860539E-02
 0.42978421E 01   0.22070814E 01  -0.89925984E 00   0.50196716E-02   0.27169866E-02   0.29926433E-02
-0.18688497E 01  -0.89926387E 00   0.46612856E 00  -0.20943384E-02  -0.11358077E-02  -0.11542848E-02
 0.10649726E-01   0.50196433E-02  -0.20943393E-02   0.12150148E-04   0.68390469E-05   0.72350127E-05
 0.61782872E-02   0.27169785E-02  -0.11358093E-02   0.68390793E-05   0.42105056E-05   0.40998587E-05
 0.62860451E-02   0.29926377E-02  -0.11542919E-02   0.72350081E-05   0.40998573E-05   0.44396959E-05

THE TRAJECTORY HAS BEEN RECTIFIED

POSITICN VECTOR (R DATE)  =     0.47887496E 04   0.30007676E 04   0.47592227E 04
VELOCITY VECTOR (V DATE)  =    -0.46795872E 01  -0.12346685E 01   0.56333339E 01

ELLIPTIC ORBIT ELEMENTS

SEMIMAJOR AXIS      (KM)  =     0.75573736E 04
ECCENTRICITY              =     0.25716597E-01
TRUE ANCMOLY DATE   (DEG) =     C.30280804E 02
R ASC OF ASC NODE   (DEG) =     0.24823674E 02
ARG OF PERIAPSE     (DEG) =     C.10362426E 02
ORBIT INCLINATION   (DEG) =     C.81478908E 02
DEL NODE PER REV    (RAD) =    -0.10779666E-02
DEL APSE PER REV    (RAD) =    -0.32381995E-02
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS           ,   0.56705973E 05   SEC (U.T.)
R VECTOR (1950)        =   0.47578092E 04   0.29717348E 04     0.48111221E 04
V VECTOR (1950)        =  -0.47240689E 01  -0.12489586E 01     0.55911060E 01
R VECTOR (DATE)        =   0.47408373E 04   0.29875642E 04     0.48180673E 04
V VECTOR (DATE)        =  -0.47279914E 01  -0.12648392E 01     0.55842160E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.27201514E 03   0.16766192E 04     0.47443488E 03
    RELATIVE VELOCITY  =  -0.46284733E 01  -0.15748302E 01     0.55911060E 01
    R, RDOT, AZ, ELEV  =   0.17635568E 04  -0.70697451E 00     0.95808168E 02   0.29152541E 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS           ,   0.56715984E 05   SEC (U.T.)
R VECTOR (1950)        =   0.47102809E 04   0.29590845E 04     0.48668571E 04
V VECTOR (1950)        =  -0.47708053E 01  -0.12782344E 01     0.55432216E 01
R VECTOR (DATE)        =   0.46932706E 04   0.29747541E 04     0.48737329E 04
V VECTOR (DATE)        =  -0.47745599E 01  -0.12942705E 01     0.55362634E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.22540619E 03   0.16605487E 04     0.53010052E 03
    RELATIVE VELOCITY  =  -0.46749719E 01  -0.16040362E 01     0.55432216E 01
    R, RDOT, AZ, ELEV  =   0.17576224E 04  -0.44314589E 00     0.93081396E 02   0.29373246E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0    ) =    0.55950000E 04   0.73343500E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =    0.56715984E 05   0.55950000E 04 (DAYS  U.T.)

POSITICN VECTOR ( DATE )  =    0.47103877E 04   0.29827978E 04   0.48703337E 04
VELOCITY VECTOR ( DATE )  =   -0.47555763E 01  -0.12835999E 01   0.55476630E 01
DELTA RADIUS              =    0.17117107E 02   0.80436883E 01  -0.33991290E 01
DELTA VELOCITY            =    0.18983539E-01   0.10670638E-01   0.11399619E-01

THE COVARIANCE MATRIX FOR THE ERRORS IN STATE  ** P **

 0.76151541E 01   0.34430392E 01  -0.13242446E 01   0.84001765E-02   0.49473142E-02   0.51929877E-02
 0.34430496E 01   0.18108737E 01  -0.64600569E 00   0.39910588E-02   0.21564669E-02   0.24949615E-02
-0.13242427E 01  -0.64600978E 00   0.33103724E 00  -0.14524485E-02  -0.77915903E-03  -0.81461810E-03
 0.84001675E-02   0.39910306E-02  -0.14524497E-02   0.94671811E-05   0.53644650E-05   0.59127972E-05
 0.49473152E-02   0.21564588E-02  -0.77916079E-03   0.53644965E-05   0.34094647E-05   0.33800687E-05
 0.51929791E-02   0.24949558E-02  -0.81462529E-03   0.59127923E-05   0.33800673E-05   0.38190292E-05

THE TRAJECTORY HAS BEEN RECTIFIED

POSITICN VECTOR (R DATE)  =    0.47103877E 04   0.29827978E 04   0.48703337E 04
VELOCITY VECTOR (V DATE)  =   -0.47555763E 01  -0.12835999E 01   0.55476630E 01

ELLIPTIC ORBIT ELEMENTS

SEMIMAJOR AXIS       (KM)  =   0.75713231E 04
ECCENTRICITY               =   0.26469280E-01
TRUE ANOMOLY DATE   (DEG)  =   0.33723414E 02
R ASC OF ASC NODE   (DEG)  =   0.24821649E 02
ARG OF PERIAPSE     (DEG)  =   0.79764781E 01
ORBIT INCLINATION   (DEG)  =   0.81477331E 02
DEL NODE PER REV    (RAD)  =  -0.10742798E-02
DEL APSE PER REV    (RAD)  =  -0.32263859E-02
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,    0.56725975E 05   SEC (U.T.)
R VECTOR (1950)        =   0.46788661E 04   0.29535597E 04     0.49213370E 04
V VECTOR (1950)        =  -0.47990211E 01  -0.12971116E 01     0.55047247E 01
R VECTOR (DATE)        =   0.46617953E 04   0.29691235E 04     0.49281669E 04
V VECTOR (DATE)        =  -0.48026562E 01  -0.13132415E 01     0.54977254E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.19488959E 03   0.16516628E 04     0.58453455E 03
    RELATIVE VELOCITY  =  -0.47029503E 01  -0.16228435E 01     0.55047247E 01
    R. RDOT. AZ. ELEV  =   0.17628535E 04  -0.21513110E 00     0.90811496E 02   0.29834409E 02

TIME FRCM JD 2433282.5 =   0.55950000E 04   DAYS          ,    0.56735965E 05   SEC (U.T.)
R VECTOR (1950)        =   0.46306926E 04   0.29404565E 04     0.49760909E 04
V VECTOR (1950)        =  -0.48445988E 01  -0.13259678E 01     0.54561558E 01
R VECTOR (DATE)        =   0.46135862E 04   0.29558583E 04     0.49828506E 04
V VECTOR (DATE)        =  -0.48480664E 01  -0.13422492E 01     0.54490899E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.14764172E 03   0.16351431E 04     0.63921820E 03
    RELATIVE VELOCITY  =  -0.47482907E 01  -0.16516296E 01     0.54561558E 01
    R, RDOT, AZ, ELEV  =   0.17618430E 04   0.48799843E-01     0.88061100E 02   0.29920722E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0    ) =    0.55950000E 04   0.73366627E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =    0.56735965E 05   0.55950000E 04 (DAYS  U.T.)

POSITICN VECTOR ( DATE )   =    0.46050422E 04   0.29518284E 04   0.49843626E 04
VELOCITY VECTOR ( DATE )   =   -0.48573963E 01  -0.13475534E 01   0.54432266E 01
DELTA RADIUS               =   -0.85440469E 01  -0.40298347E 01   0.15120695E 01
DELTA VELOCITY             =   -0.93299133E-02  -0.53041209E-02  -0.58632712E-02


THE COVARIANCE MATRIX FOR THE ERRORS IN STATE  ** P **


 0.62860913E 01   0.28357921E 01  -0.94783141E 00   0.68164854E-02   0.40802592E-02   0.44082788E-02
 0.28358024E 01   0.15261286E 01  -0.46895583E 00   0.32626860E-02   0.17603138E-02   0.21356378E-02
-0.94782928E 00  -0.46895995E 00   0.24638392E 00  -0.10132225E-02  -0.53345071E-03  -0.57087202E-03
 0.68164777E-02   0.32626583E-02  -0.10132242E-02   0.75900771E-05   0.43305428E-05   0.49655848E-05
 0.40802606E-02   0.17603059E-02  -0.53345268E-03   0.43305730E-05   0.28492756E-05   0.28652269E-05
 0.44082702E-02   0.21356323E-02  -0.57087944E-03   0.49655795E-05   0.28652256E-05   0.33695671E-05


THE TRAJECTORY HAS BEEN RECTIFIED

POSITION VECTOR (R DATE)   =    0.46050422E 04   0.29518284E 04   0.49843626E 04
VELOCITY VECTOR (V DATE)   =   -0.48573963E 01  -0.13475534E 01   0.54432266E 01


ELLIPTIC ORBIT ELEMENTS

SEMIMAJOR AXIS       (KM)  =    0.75653608E 04
ECCENTRICITY               =    0.26087942E-01
TRUE ANOMOLY DATE    (DEG) =    0.34043023E 02
R ASC OF ASC NODE    (DEG) =    0.24814227E 02
ARG OF PERIAPSE      (DEG) =    0.88831836E 01
ORBIT INCLINATION    (DEG) =    0.81480377E 02
DEL NODE PER REV     (RAD) =   -0.10755490E-02
DEL APSE PER REV     (RAD) =   -0.32316264E-02
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS           ,   0.56745977E 05   SEC (U.T.)
R VECTOR (1950)        =   0.45724467E 04   0.29222744E 04   0.50345444E 04
V VECTOR (1950)        =  -0.49001309E 01  -0.13603957E 01   0.53990615E 01
R VECTOR (DATE)        =   0.45553165E 04   0.29374807E 04   0.50412192E 04
V VECTOR (DATE)        =  -0.49034000E 01  -0.13768619E 01   0.53919145E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.90336914E 02   0.16135071E 04   0.69758679E 03
    RELATIVE VELOCITY  =  -0.48035852E 01  -0.16859871E 01   0.53990615E 01
    R, RDOT, AZ, ELEV  =   0.17601685E 04   0.34770581E 00   0.84918569E 02   0.29810560E 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS           ,   0.56755988E 05   SEC (U.T.)
R VECTOR (1950)        =   0.45231660E 04   0.29085117E 04   0.50883471E 04
V VECTOR (1950)        =  -0.49447980E 01  -0.13890298E 01   0.53492337E 01
R VECTOR (DATE)        =   0.45060040E 04   0.29235523E 04   0.50949500E 04
V VECTOR (DATE)        =  -0.49478985E 01  -0.14056446E 01   0.53420215E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   0.41991699E 02   0.15963211E 04   0.75131763E 03
    RELATIVE VELOCITY  =  -0.48480147E 01  -0.17145507E 01   0.53492337E 01
    R, RDOT, AZ, ELEV  =   0.17647897E 04   0.61107774E 00   0.82170632E 02   0.29733447E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0     ) =    0.55950000E 04    0.73389801E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =    0.56755988E 05    0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )  =    0.45039788E 04    0.29225968E 04    0.50952590E 04
VELOCITY VECTOR ( DATE )  =   -0.49500748E 01   -0.14069025E 01    0.53405868E 01
DELTA RADIUS              =   -0.20252047E 01   -0.95555549E 00    0.30897555E 00
DELTA VELOCITY            =   -0.21762910E-02   -0.12579913E-02   -0.14347029E-02
```

THE COVARIANCE MATRIX FOR THE ERRORS IN STATE ** P **

0.52892697E 01	0.23779263E 01	-0.67438260E 00	0.56345648E-02	0.34326908E-02	0.38096787E-02
0.23779363E 01	0.13085026E 01	-0.33869328E 00	0.27153228E-02	0.14632861E-02	0.18594624E-02
-0.67438018E 00	-0.33869741E 00	0.19241503E 00	-0.69749247E-03	-0.35536857E-03	-0.38634342E-03
0.56345585E-02	0.27152958E-02	-0.69749461E-03	0.61984103E-05	0.35619463E-05	0.42447113E-05
0.34326923E-02	0.14632783E-02	-0.35537078E-03	0.35619751E-05	0.24340928E-05	0.24740744E-05
0.38096704E-02	0.18594573E-02	-0.38635114E-03	0.42447057E-05	0.24740734E-05	0.30228014E-05

THE TRAJECTORY HAS BEEN RECTIFIED

```
POSITION VECTOR (R DATE)  =    0.45039788E 04    0.29225968E 04    0.50952590E 04
VELOCITY VECTOR (V DATE)  =   -0.49500748E 01   -0.14069025E 01    0.53405868E 01
```

ELLIPTIC ORBIT ELEMENTS

```
SEMIMAJOR AXIS        (KM)  =    0.75640149E 04
ECCENTRICITY                =    0.25958102E-01
TRUE ANOMOLY DATE     (DEG) =    0.35226962E 02
R ASC OF ASC NODE     (DEG) =    0.24807950E 02
ARG OF PERIAPSE       (DEG) =    0.88834790E 01
ORBIT INCLINATION     (DEG) =    0.81482015E 02
DEL NODE PER REV      (RAD) =   -0.10757120E-02
DEL APSE PER REV      (RAD) =   -0.32328847E-02
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS         ,   0.56765979E 05   SEC (U.T.)
R VECTOR (1950)        =   0.44706077E 04   0.28928168E 04   0.51444072E 04
V VECTOR (1950)        =  -0.49918471E 01  -0.14189453E 01   0.52955181E 01
R VECTOR (DATE)        =   0.44534170E 04   0.29076808E 04   0.51509333E 04
V VECTOR (DATE)        =  -0.49947691E 01  -0.14357165E 01   0.52882372E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -C.96271362E 01   0.15771972E 04   0.80730096E 03
    RELATIVE VELOCITY  =  -0.48948267E 01  -0.17443956E 01   0.52955181E 01
    R. RDOT, AZ, ELEV  =   0.17718291E 04   0.88662280E 00   0.79289180E 02   0.29569404E 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS         ,   0.56775969E 05   SEC (U.T.)
R VECTOR (1950)        =   0.44205178E 04   0.28784993E 04   0.51970599E 04
V VECTOR (1950)        =  -0.50353837E 01  -0.14472054E 01   0.52447619E 01
R VECTOR (DATE)        =   0.44032988E 04   0.28931950E 04   0.52035131E 04
V VECTOR (DATE)        =  -0.50381370E 01  -0.14641213E 01   0.52374175E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.58774597E 02   0.15594597E 04   0.85988068E 03
    RELATIVE VELOCITY  =  -0.49381261E 01  -0.17725849E 01   0.52447619E 01
    R, RDOT, AZ, ELEV  =   0.17817867E 04   0.11425785E 01   0.76609653E 02   0.29342310E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0    ) =    0.55950000E 04    0.73412928E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =    0.56775969E 05    0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )  =    0.44016112E 04    0.28923944E 04    0.52037346E 04
VELOCITY VECTOR ( DATE )  =   -0.50399230E 01   -0.14651641E 01    0.52361878E 01
DELTA RADIUS              =   -0.16875151E 01   -0.80059736E 00    0.22155910E 00
DELTA VELOCITY            =   -0.17859844E-02   -0.10428051E-02   -0.12296080E-02


THE COVARIANCE MATRIX FOR THE ERRORS IN STATE  ** P **


 0.45616083E 01   0.20428281E 01  -0.47639907E 00   0.47717908E-02   0.29608230E-02   0.33696339E-02
 0.20428377E 01   0.11469209E 01  -0.24311736E 00   0.23137061E-02   0.12467744E-02   0.16553338E-02
-0.47639635E 00  -0.24312149E 00   0.15942915E 00  -0.47110161E-03  -0.22645126E-03  -0.24678409E-03
 0.47717859E-02   0.23136800E-02  -0.47110421E-03   0.51854158E-05   0.30018262E-05   0.37122673E-05
 0.29608246E-02   0.12467668E-02  -0.22645373E-03   0.30018535E-05   0.21336416E-05   0.21868719E-05
 0.33696257E-02   0.16553291E-02  -0.24679213E-03   0.37122613E-05   0.21868712E-05   0.27688954E-05


THE TRAJECTORY HAS BEEN RECTIFIED

POSITION VECTOR (R DATE)  =    0.44016112E 04    0.28923944E 04    0.52037346E 04
VELOCITY VECTOR (V DATE)  =   -0.50399230E 01   -0.14651641E 01    0.52361878E 01


ELLIPTIC ORBIT ELEMENTS

SEMIMAJOR AXIS       (KM)  =    0.75625949E 04
ECCENTRICITY               =    C.25812171E-01
TRUE ANOMOLY DATE    (DEG) =    0.36527850E 02
R ASC OF ASC NODE    (DEG) =    0.24801555E 02
ARG OF PERIAPSE      (DEG) =    0.87612374E 01
ORBIT INCLINATION    (DEG) =    0.81483406E 02
DEL NODE PER REV     (RAD) =   -0.10759252E-02
DEL APSE PER REV     (RAD) =   -0.32341792E-02
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS           ,   0.56785980E 05   SEC (U.T.)
R VECTOR (1950)        =   0.43672844E 04   0.28623478E 04   0.52520501E 04
V VECTOR (1950)        =  -0.50808895E 01  -0.14765157E 01   0.51900483E 01
R VECTOR (DATE)        =   0.43500398E 04   0.28768648E 04   0.52584256E 04
V VECTOR (DATE)        =  -0.50834649E 01  -0.14935828E 01   0.51826375E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.11105920E 03   0.15398739E 04   0.91479321E 03
    RELATIVE VELOCITY  =  -0.49833946E 01  -0.18018241E 01   0.51900483E 01
    R, RDOT, AZ, ELEV  =   0.17945451E 04   0.14079840E 01   0.73824601E 02   0.29032542E 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS           ,   0.56795992E 05   SEC (U.T.)
R VECTOR (1950)        =   0.43162046E 04   0.28474258E 04   0.53037499E 04
V VECTOR (1950)        =  -0.51234562E 01  -0.15045051E 01   0.51381832E 01
R VECTOR (DATE)        =   0.42989351E 04   0.28617711E 04   0.53100510E 04
V VECTOR (DATE)        =  -0.51258622E 01  -0.15217138E 01   0.51307102E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.16118573E 03   0.15215206E 04   0.96641858E 03
    RELATIVE VELOCITY  =  -0.50257235E 01  -0.18297423E 01   0.51381832E 01
    R, RDOT, AZ, ELEV  =   0.18096880E 04   0.16531689E 01   0.71247322E 02   0.28669081E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0    ) =    0.55950000E 04   0.73436102E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =     0.56795992E 05   0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )   =    0.42975869E 04    0.28611278E 04    0.53101990E 04
VELOCITY VECTOR ( DATE )   =   -0.51272674E 01   -0.15225428E 01    0.51297004E 01
DELTA RADIUS               =   -0.13481887E 01   -0.64331163E 00    0.14807598E 00
DELTA VELOCITY             =   -0.14051696E-02   -0.82896966E-03   -0.10098147E-02


THE COVARIANCE MATRIX FOR THE ERRORS IN STATE  ** P **


 0.40321047E 01   0.17990828E 01  -0.32964701E 00   0.41402766E-02   0.26170461E-02   0.30506968E-02
 0.17990920E 01   0.10275135E 01  -0.17127345E 00   0.20187490E-02   0.10896517E-02   0.15069055E-02
-0.32964399E 00  -0.17127753E 00   0.14020236E 00  -0.30490845E-03  -0.13071795E-03  -0.13834314E-03
 0.41402730E-02   0.20187238E-02  -0.30491143E-03   0.44430394E-05   0.25914782E-05   0.33210223E-05
 0.26170478E-02   0.10896443E-02  -0.13072068E-03   0.25915041E-05   0.19160846E-05   0.19782208E-05
 0.30506888E-02   0.15069012E-02  -0.13835148E-03   0.33210158E-05   0.19782204E-05   0.25891506E-05


THE TRAJECTORY HAS BEEN RECTIFIED

POSITION VECTOR (R DATE)   =    0.42975869E 04    0.28611278E 04    0.53101990E 04
VELOCITY VECTOR (V DATE)   =   -0.51272674E 01   -0.15225428E 01    0.51297004E 01


ELLIPTIC ORBIT ELEMENTS

SEMIMAJOR AXIS      (KM)  =    0.75610950E 04
ECCENTRICITY              =    0.25651617E-01
TRUE ANOMOLY DATE   (DEG) =    0.37953820E 02
R ASC OF ASC NODE   (DEG) =    0.24795023E 02
ARG OF PERIAPSE     (DEG) =    0.85129178E 01
ORBIT INCLINATION   (DEG) =    0.81484549E 02
DEL NODE PER REV    (RAD) =   -0.10761910E-02
DEL APSE PER REV    (RAD) =   -0.32355151E-02
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,    0.56805982E 05   SEC (U.T.)
R VECTOR (1950)        =   0.42625364E 04   0.28308924E 04     0.53574481E 04
V VECTOR (1950)        =  -0.51672257E 01  -0.15330768E 01     0.50827404E 01
R VECTOR (DATE)        =   0.42452444E 04   0.28450575E 04     0.53636708E 04
V VECTOR (DATE)        =  -0.51694552E 01  -0.15504309E 01     0.50752035E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.21389874E 03   0.15015574E 04     0.10200385E 04
    RELATIVE VELOCITY  =  -0.50692561E 01  -0.18582426E 01     0.50827404E 01
    R, RDOT, AZ, ELEV  =   0.18278145E 04   0.19031691E 01     0.68610835E 02   0.28234589E 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,    0.56815973E 05   SEC (U.T.)
R VECTOR (1950)        =   0.42107049E 04   0.28154380E 04     0.54079653E 04
V VECTOR (1950)        =  -0.52086313E 01  -0.15606683E 01     0.50300118E 01
R VECTOR (DATE)        =   0.41933915E 04   0.28294290E 04     0.54141124E 04
V VECTOR (DATE)        =  -0.52106915E 01  -0.15781600E 01     0.50224145E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.26477148E 03   0.14826803E 04     0.10704800E 04
    RELATIVE VELOCITY  =  -0.51104248E 01  -0.18857627E 01     0.50300118E 01
    R, RDOT, AZ, ELEV  =   0.18478020E 04   0.21331498E 01     0.66180216E 02   0.27756544E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0     ) =    0.55950000E 04    0.73459229E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =      0.56815973E 05    0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )  =      0.41922181E 04    0.28288652E 04    0.54142167E 04
VELOCITY VECTOR ( DATE )  =     -0.52118960E 01    0.15788771E 01    0.50215121E 01
DELTA RADIUS              =     -0.11733451E 01   -0.56377818E 00    0.10433077E 00
DELTA VELOCITY            =     -0.12045291E-02   -0.71705695E-03   -0.90237358E-03


THE COVARIANCE MATRIX FOR THE ERRORS IN STATE   ** P **


 0.36462679E 01   0.16221735E 01  -0.21765470E 00   0.36744718E-02   0.23654449E-02   0.28220602E-02
 0.16221823E 01   0.93931942E 00  -0.11564752E 00   0.18008752E-02   0.97571689E-03   0.14004585E-02
-0.21765140E 00  -0.11565154E 00   0.13015359E 00  -0.17946652E-03  -0.57444988E-04   0.51272011E-04
 0.36744694E-02   0.18008509E-02  -0.17946981E-03   0.38927084E-05   0.22878058E-05   0.30335407E-05
 0.23654466E-02   0.97570964E-03  -0.57447978E-04   0.22878303E-05   0.17577487E-05   0.18276804E-05
 0.28220523E-02   0.14004547E-02  -0.51280655E-04   0.30335337E-05   0.18276802E-05   0.24666377E-05


THE TRAJECTORY HAS BEEN RECTIFIED

POSITICN VECTOR (R DATE)  =      0.41922181E 04    0.28288652E 04    0.54142167E 04
VELOCITY VECTOR (V DATE)  =     -0.52118960E 01   -0.15788771E 01    0.50215121E 01


ELLIPTIC ORBIT ELEMENTS

SEMIMAJOR AXIS       (KM)  =     0.75593933E 04
ECCENTRICITY               =     0.25472217E-01
TRUE ANOMOLY DATE    (DEG) =     0.39486772E 02
R ASC OF ASC NODE    (DEG) =     0.24788316E 02
ARG OF PERIAPSE      (DEG) =     0.81528981E 01
ORBIT INCLINATION    (DEG) =     0.81485473E 02
DEL NODE PER REV     (RAD) =    -0.10765397E-02
DEL APSE PER REV     (RAD) =    -C.32369986E-02
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS           ,    0.56825984E 05   SEC (U.T.)
R VECTOR (1950)        =   0.41562554E 04   0.27983954E 04    0.54605928E 04
V VECTOR (1950)        =  -0.52509974E 01  -0.15887006E 01    0.49735554E 01
R VECTOR (DATE)        =   0.41389229E 04   0.28122036E 04    0.54666605E 04
V VECTOR (DATE)        =  -0.52528815E 01  -0.16063330E 01    0.49658963E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.31825616E 03   0.14622020E 04    0.11230281E 04
    RELATIVE VELOCITY  =  -0.51525537E 01  -0.19137232E 01    0.49735554E 01
    R, RDOT, AZ, ELEV  =   0.18709660E 04   0.23661706E 01    0.63705619E 02   0.27215741E 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS           ,    0.56835996E 05   SEC (U.T.)
R VECTOR (1950)        =   0.41034835E 04   0.27823537E 04    0.55101157E 04
V VECTOR (1950)        =  -0.52913960E 01  -0.16159961E 01    0.49197735E 01
R VECTOR (DATE)        =   0.40861329E 04   0.27959847E 04    0.55161064E 04
V VECTOR (DATE)        =  -0.52931103E 01  -0.16337627E 01    0.49120553E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.37005975E 03   0.14427311E 04    0.11724741E 04
    RELATIVE VELOCITY  =  -0.51927152E 01  -0.19409468E 01    0.49197735E 01
    R, RDOT, AZ, ELEV  =   0.18955508E 04   0.25795443E 01    0.61433715E 02   0.26644004E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0    ) =   0.55950000E 04   0.73482403E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =   0.56835996E 05   0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )  =   0.40851295E 04   0.27954996E 04   0.55161741E 04
VELOCITY VECTOR ( DATE )  =  -0.52941244E 01  -0.16343726E 01   0.49112634E 01
DELTA RADIUS              =  -0.10033764E 01  -0.48511968E 00   0.67740413E-01
DELTA VELOCITY            =  -0.10142095E-02  -0.60984703E-03  -0.79188447E-03
```

THE COVARIANCE MATRIX FOR THE ERRORS IN STATE ** P **

0.33657141E 01	0.14947918E 01	-0.12879043E 00	0.33281316E-02	0.21807105E-02	0.26619730E-02
0.14948003E 01	0.87454312E 00	-0.70858383E-01	0.16391146E-02	0.89352528E-03	0.13263095E-02
-0.12878688E 00	-0.70862310E-01	0.12635755E 00	-0.81226991E-04	0.89940090E-06	0.21636019E-04
0.33281303E-02	0.16390910E-02	-0.81230513E-04	0.34791018E-05	0.20604314E-05	0.28231382E-05
0.21807122E-02	0.89351818E-03	0.89615719E-06	0.20604546E-05	0.16419950E-05	0.17207560E-05
0.26619653E-02	0.13263061E-02	0.21627089E-04	0.28231309E-05	0.17207561E-05	0.23896263E-05

THE TRAJECTORY HAS BEEN RECTIFIED

```
POSITION VECTOR (R DATE)  =   0.40851295E 04   0.27954996E 04   0.55161741E 04
VELOCITY VECTOR (V DATE)  =  -0.52941244E 01  -0.16343726E 01   0.49112634E 01
```

ELLIPTIC ORBIT ELEMENTS

```
SEMIMAJOR AXIS       (KM)  =   0.75574844E 04
ECCENTRICITY               =   0.25276957E-01
TRUE ANOMOLY DATE    (DEG) =   0.41138099E 02
R ASC OF ASC NODE    (DEG) =   0.24781423E 02
ARG OF PERIAPSE      (DEG) =   0.76744313E 01
ORBIT INCLINATION    (DEG) =   0.81486180E 02
DEL NODE PER REV     (RAD) =  -0.10769735E-02
DEL APSE PER REV     (RAD) =  -0.32386357E-02
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,    0.56845986E 05   SEC (U.T.)
R VECTOR (1950)        =   0.40484927E 04   0.27648779E 04    0.55514535E 04
V VECTOR (1950)        =  -0.53321789E 01  -0.16433635E 01    0.48625682E 01
R VECTOR (DATE)        =   0.40311263E 04   0.27783242E 04    0.55673640E 04
V VECTOR (DATE)        =  -C.53337181E 01  -0.16612656E 01    0.48547905E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.42407928E 03   0.14218239E 04    0.12237316E 04
    RELATIVE VELOCITY  =  -0.52332613E 01  -0.19682422E 01    0.48625682E 01
    R, RDOT, AZ, ELEV  =   0.19232643E C4   0.27928050E 01    0.59147023E 02   0.26023618E 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,    0.56855977E 05   SEC (U.T.)
R VECTOR (1950)        =   0.39950242E 04   0.27483252E 04    0.56097617E 04
V VECTOR (1950)        =  -0.53713904E 01  -0.16702398E 01    0.48079840E 01
R VECTOR (DATE)        =   0.39776433E 04   0.27615920E 04    0.56155942E 04
V VECTOR (DATE)        =  -0.53727600E 01  -0.16882721E 01    0.48001491E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.47657278E 03   0.14018460E 04    0.12719618F 04
    RELATIVE VELOCITY  =  -0.52722361F 01  -0.19950463E 01    0.48079840E 01
    R, RDOT, AZ, ELFV  =   0.19519684E 04   0.29874605E 01    0.57053509E 02   0.25383678E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0     ) =   0.55950000E 04   0.73505530E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =     0.56855977E 05   0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )    =   0.39768107E 04   0.27611872E 04   0.56156322E 04
VELOCITY VECTOR ( DATE )    =  -0.53735884E 01  -0.16887757E 01   0.47994752E 01
DELTA RADIUS                =  -0.83256809E 00  -0.40479624E 00   0.38078536E-01
DELTA VELOCITY              =  -0.82838651E-03  -0.50365795E-03  -0.67387668E-03
```

THE COVARIANCE MATRIX FOR THE ERRORS IN STATE ** P **

0.31619938E 01	0.14038908E 01	-0.55522562E-01	0.30679369E-02	0.20443851E-02	0.25532510E-02
0.14038989E 01	0.82725813E 00	-0.33398278E-01	0.15181459E-02	0.83458117E-03	0.12766683E-02
-0.55518792E-01	-0.33402103E-01	0.12700576E 00	-0.15610731E-05	0.49126507E-04	0.85144790E-04
0.30679365E-02	0.15181229E-02	-0.15647483E-05	0.31633296E-05	0.18877962E-05	0.26695006E-05
0.20443869E-02	0.83457421E-03	0.49123018E-04	0.18878183E-05	0.15568317E-05	0.16462227E-05
0.25532434E-02	0.12766653E-02	0.85135596E-04	0.26694930E-05	0.16462230E-05	0.23479941E-05

THE TRAJECTORY HAS BEEN RECTIFIED

```
POSITION VECTOR (R DATE)    =   0.39768107E 04   0.27611872E 04   0.56156322E 04
VELOCITY VECTOR (V DATE)    =  -0.53735884E 01  -0.16887757E 01   0.47994752E 01
```

ELLIPTIC ORBIT ELEMENTS

```
SEMIMAJOR AXIS        (KM)  =   0.75553696E 04
ECCENTRICITY                =   0.25067697E-01
TRUE ANOMOLY DATE     (DEG) =   0.42904823E 02
R ASC OF ASC NODE     (DEG) =   0.24774336E 02
ARG OF PERIAPSE       (DEG) =   0.70756692E 01
ORBIT INCLINATION     (DEG) =   0.81486670E 02
DEL NODE PER REV      (RAD) =  -0.10774923E-02
DEL APSE PER REV      (RAD) =  -0.32404260E-02
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS            ,   0.56865968E 05   SEC (U.T.)
R VECTOR (1950)        =   0.39394167E 04   0.27303988E 04   0.56599033E 04
V VECTOR (1950)        =  -0.54106622E 01  -0.16969849E 01   0.47499732E 01
R VECTOR (DATE)        =   0.39220232E 04   0.27434790E 04   0.56656547E 04
V VECTOR (DATE)        =  -0.54118577E 01  -0.17151475E 01   0.47420810E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.53120084E 03   0.13804877E 04   0.13220223E 04
    RELATIVE VELOCITY  =  -0.53112713E 01  -0.20217191E 01   0.47499732E 01
    R, RDOT, AZ, ELEV  =   0.19838507E 04   0.31806648E 01   0.54954942E 02   0.24706215E 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS            ,   0.56875958E 05   SEC (U.T.)
R VECTOR (1950)        =   0.38851697E 04   0.27133122E 04   0.57070821E 04
V VECTOR (1950)        =  -0.54487562E 01  -0.17234874E 01   0.46945059E 01
R VECTOR (DATE)        =   0.38677651E 04   0.27262103E 04   0.57127543E 04
V VECTOR (DATE)        =  -0.54497821E 01  -0.17417765E 01   0.46865580E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.58446460E 03   0.13599744E 04   0.13691219E 04
    RELATIVE VELOCITY  =  -0.53491287E 01  -0.20481491E 01   0.46945059E 01
    R, RDOT, AZ, ELEV  =   0.20163393E 04   0.33567252E 01   0.53036331E 02   0.24017990E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0    ) =    0.55950000E 04   0.73528656E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =    0.56875958E 05   0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )  =    0.38669553E 04   0.27258134E 04   0.57127747E 04
VELOCITY VECTOR ( DATE )  =   -0.54505754E 01  -0.17422627E 01   0.46858867E 01
DELTA RADIUS              =   -0.80973204E 00  -0.39685396E 00   0.20442979E-01
DELTA VELOCITY            =   -0.79328725E-03  -0.48619946E-03  -0.67122000E-03


THE COVARIANCE MATRIX FOR THE ERRORS IN STATE  ** P **


0.30155386E 01   0.13405050E 01   0.75314447E-02   0.28707638E-02   0.19437159E-02   0.24842127E-02
0.13405128E 01   0.79336971E 00  -0.71904744E-03   0.14273050E-02   0.79301952E-03   0.12462634E-02
0.75354029E-02  -0.72275603E-03   0.13098214E 00   0.65603963E-04   0.90658076E-04   0.14284012E-03
0.28707642E-02   0.14272826E-02   0.65600204E-04   0.29183048E-05   0.17549013E-05   0.25585377E-05
0.19437177E-02   0.79301268E-03   0.90654353E-04   0.17549222E-05   0.14939706E-05   0.15963738E-05
0.24842051E-02   0.12462609E-02   0.14283068E-03   0.25585297E-05   0.15963744E-05   0.23351274E-05


THE TRAJECTORY HAS BEEN RECTIFIED

POSITION VECTOR (R DATE)  =    0.38669553E 04   0.27258134E 04   0.57127747E 04
VELOCITY VECTOR (V DATE)  =   -0.54505754E 01  -0.17422627E 01   0.46858867E 01


ELLIPTIC ORBIT ELEMENTS

SEMIMAJOR AXIS       (KM) =    0.75529485E 04
ECCENTRICITY              =    0.24842583E-01
TRUE ANOMOLY DATE   (DEG) =    0.44781382E 02
R ASC OF ASC NODE   (DEG) =    0.24767022E 02
ARG OF PERIAPSE     (DEG) =    0.63654473E 01
ORBIT INCLINATION   (DEG) =    0.81486972E 02
DEL NODE PER REV    (RAD) =   -0.10781209E-02
DEL APSE PER REV    (RAD) =   -0.32424589E-02
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS           ,   0.56885970E 05   SEC (U.T.)
R VECTOR (1950)        =   0.38287134E 04   0.26948396E 04     0.57561172E 04
V VECTOR (1950)        =  -0.54867278E 01  -0.17497378E 01     0.46355009E 01
R VECTOR (DATE)        =   0.38112996E 04   0.27075481E 04     0.57617071E 04
V VECTOR (DATE)        =  -0.54875798E 01  -0.17681530E 01     0.46274976E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.63993198E 03   0.13380639E 04     0.14180747E 04
    RELATIVE VELOCITY  =  -0.53868635E 01  -0.20743267E 01     0.46355009E 01
    R, RDOT, AZ, ELEV  =   0.20520389E 04   0.35306986E 01     0.51115834E 02   0.23299059E 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS           ,   0.56895981E 05   SEC (U.T.)
R VECTOR (1950)        =   0.37735983E 04   0.26771913E 04     0.58022424E 04
V VECTOR (1950)        =  -0.55237656E 01  -0.17759096E 01     0.45790605E 01
R VECTOR (DATE)        =   0.37561768E 04   0.26897149E 04     0.58077520E 04
V VECTOR (DATE)        =  -0.55244477E 01  -0.17944477E 01     0.45710031E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.69405426E 03   0.13169830E 04     0.14641196E 04
    RELATIVE VELOCITY  =  -0.54236643E 01  -0.21004256E 01     0.45790605E 01
    R, RDOT, AZ, ELEV  =   0.20880138E 04   0.36888570E 01     0.49363441E 02   0.22579087E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0   ) =  0.55950000E 04   0.73551831E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =  0.56895981E 05   0.55950000E 04 (DAYS  U.T.)

POSITICN VECTOR ( DATE )  =  0.37554767E 04   0.26893699E 04   0.58077543E 04
VELOCITY VECTOR ( DATE )  = -0.55251225E 01  -0.17948661E 01   0.45704088E 01
DELTA RADIUS              = -0.70014572E 00  -0.34495654E 00   0.23362258E-02
DELTA VELOCITY            = -0.67480320E-03  -0.41840800E-03  -0.59430349E-03


THE COVARIANCE MATRIX FOR THE ERRORS IN STATE  ** P **


0.29123604E 01  0.12981846E 01  0.64098996E-01  0.27201645E-02  0.18696148E-02  0.24463613E-02
0.12981922E 01  0.76991492E 00  0.28970017E-01  0.13589457E-02  0.76463254E-03  0.12312467E-02
0.64103110E-01  0.28966438E-01  0.13763373E 00  0.12437948E-03  0.12783574E-03  0.19732716E-03
0.27201656E-02  0.13589238E-02  0.12437570E-03  0.27249171E-05  0.16511290E-05  0.24799969E-05
0.18696167E-02  0.76462581E-03  0.12783179E-03  0.16511489E-05  0.14475297E-05  0.15656066E-05
0.24463538E-02  0.12312445E-02  0.19731750E-03  0.24799886E-05  0.15656073E-05  0.23461889E-05


THE TRAJECTORY HAS BEEN RECTIFIED

POSITION VECTOR (R DATE)  =  0.37554767E 04   0.26893699E 04   0.58077543E 04
VELOCITY VECTOR (V DATE)  = -0.55251225E 01  -0.17948661E 01   0.45704088E 01


ELLIPTIC ORBIT ELEMENTS

SEMIMAJOR AXIS       (KM)  =  0.75502838E 04
ECCENTRICITY               =  0.24609508E-01
TRUE ANOMOLY DATE    (DEG) =  0.46786610E 02
R ASC OF ASC NODE    (DEG) =  0.24759491E 02
ARG OF PERIAPSE      (DEG) =  0.55266028E 01
ORBIT INCLINATION    (DEG) =  0.81487071E 02
DEL NODE PER REV     (RAD) = -0.10788447E-02
DEL APSE PER REV     (RAD) = -0.32446827E-02
```

```
TIME FROM JD 2433282.5 =    0.55950000E 04   DAYS          ,   0.56905972E 05   SEC (U.T.)
R VECTOR (1950)        =    0.37166350E 04    0.26583001E 04   0.58499607E 04
V VECTOR (1950)        =   -0.55601816E 01   -0.18014918E 01   0.45194098E 01
R VECTOR (DATE)        =    0.36992077E 04    0.26706325E 04   0.58553871E 04
V VECTOR (DATE)        =   -0.55606911E 01   -0.18201507E 01   0.45112993E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   -0.75002191E 03    0.12946583E 04   0.15117547E 04
    RELATIVE VELOCITY  =   -0.54598439E 01   -0.21259347E 01   0.45194098E 01
    R, RDOT, AZ, ELEV  =    0.21269873E 04    0.38434091E 01   0.47622239E 02   0.21840970E 02

TIME FROM JD 2433282.5 =    0.55950000E 04   DAYS          ,   0.56915962E 05   SEC (U.T.)
R VECTOR (1950)        =    0.36609055E 04    0.26401732E 04   0.58948276E 04
V VECTOR (1950)        =   -0.55959995E 01   -0.18272156E 01   0.44622586E 01
R VECTOR (DATE)        =    0.36434740E 04    0.26523186E 04   0.59001727E 04
V VECTOR (DATE)        =   -0.55963396E 01   -0.18459934E 01   0.44540957E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =   -0.80475186E 03    0.12731028E 04   0.155[illegible]E 04
    RELATIVE VELOCITY  =   -0.54954255E 01   -0.21515854E 01   0.44622586E 01
    R, RDOT, AZ, ELEV  =    0.21659257E 04    0.39839547E 01   0.46033250E 02   0.21107064E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0    ) =   0.55950000E 04   0.73574957E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =   0.56915962E 05   0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )  =   0.36428143E 04   0.26519911E 04   0.59001613E 04
VELOCITY VECTOR ( DATE )  =  -0.55969653E 01  -0.18463847E 01   0.44535230E 01
DELTA RADIUS              =  -0.65971595E 00  -0.32748242E 00  -0.11443097E-01
DELTA VELOCITY            =  -0.62566282E-03  -0.39139928E-03  -0.57272184E-03


THE COVARIANCE MATRIX FOR THE ERRORS IN STATE  ** P **


0.28416660E 01   0.12718376E 01   0.11649307E 00   0.26039345E-02   0.18152158E-02   0.24325936E-02
0.12718450E 01   0.75453749E 00   0.56785070E-01   0.13072923E-02   0.74615302E-03   0.12283239E-02
0.11649731E 00   0.56781634E-01   0.14654554E 00   0.17726610E-03   0.16208031E-03   0.25022358E-03
0.26039363E-02   0.13072710E-02   0.17726235E-03   0.25694359E-05   0.15687529E-05   0.24257481E-05
0.18152177E-02   0.74614639E-03   0.16207614E-03   0.15687719E-05   0.14131185E-05   0.15493522E-05
0.24325862E-02   0.12283221E-02   0.25021370E-03   0.24257397E-05   0.15493531E-05   0.23766278E-05


THE TRAJECTORY HAS BEEN RECTIFIED

POSITION VECTOR (R DATE)  =   0.36428143E 04   0.26519911E 04   0.59001613E 04
VELOCITY VECTOR (V DATE)  =  -0.55969653E 01  -0.18463847E 01   0.44535230E 01


ELLIPTIC ORBIT ELEMENTS

SEMIMAJOR AXIS       (KM)  =   0.75473341E 04
ECCENTRICITY               =   0.24369313E-01
TRUE ANOMOLY DATE   (DEG)  =   0.48912405E 02
R ASC OF ASC NODE   (DEG)  =   0.24751719E 02
ARG OF PERIAPSE     (DEG)  =   0.45629190E 01
ORBIT INCLINATION   (DEG)  =   0.81486983E 02
DEL NODE PER REV    (RAD)  =  -0.10796738E-02
DEL APSE PER REV    (RAD)  =  -0.32471347E-02
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,   0.56925974E 05   SEC (U.T.)
R VECTOR (1950)        =   0.36031685E 04   0.26207690E 04   0.59414060E 04
V VECTOR (1950)        =  -0.56310672E 01  -0.18522642E 01   0.44017177E 01
R VECTOR (DATE)        =   0.35857346E 04   0.26327207E 04   0.59466669E 04
V VECTOR (DATE)        =  -0.56312351E 01  -0.18711583E 01   0.43935037E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.86148370E 03   0.12502594E 04   0.16030345E 04
    RELATIVE VELOCITY  =  -0.55302565E 01  -0.21765605E 01   0.44017177E 01
    R, RDOT, AZ, ELEV  =   0.22079453E 04   0.41210558E 01   0.44450987E 02   0.20358755E 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,   0.56935985E 05   SEC (U.T.)
R VECTOR (1950)        =   0.35466199E 04   0.26020983E 04   0.59851826E 04
V VECTOR (1950)        =  -0.56658001E 01  -0.18776361E 01   0.43436475E 01
R VECTOR (DATE)        =   0.35291851E 04   0.26138602E 04   0.59903610E 04
V VECTOR (DATE)        =  -0.56657985E 01  -0.18966453E 01   0.43353828E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.91702179E 03   0.12281496E 04   0.16467286E 04
    RELATIVE VELOCITY  =  -0.55647525E 01  -0.22018586E 01   0.43436475E 01
    R, RDOT, AZ, ELEV  =   0.22496656E 04   0.42457848E 01   0.43006824E 02   0.19619360E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0    ) =    0.55950000E 04   0.73598132E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =    0.56935985E 05   0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )  =    0.35286039E 04   0.26135703E 04   0.59903381E 04
VELOCITY VECTOR ( DATE )  =   -0.56663404E 01  -0.18969886E 01   0.43348668E 01
DELTA RADIUS              =   -0.58125498E 00  -0.28988180E 00  -0.22928023E-01
DELTA VELOCITY            =   -0.54191070E-03  -0.34333030E-03  -0.51598955E-03

THE COVARIANCE MATRIX FOR THE ERRORS IN STATE  ** P **

0.27964006E 01   0.12582067E 01   0.16670163E 00   0.25136543E-02   0.17759021E-02   0.24386754E-02
0.12582139E 01   0.74572824E 00   0.83717056E-01   0.12683913E-02   0.73543926E-03   0.12355961E-02
0.16670597E 00   0.83713772E-01   0.15755039E 00   0.22633259E-03   0.19461011E-03   0.30309571E-03
0.25136567E-02   0.12683704E-02   0.22632891E-03   0.24420849E-05   0.15023385E-05   0.23904566E-05
0.17759041E-02   0.73543272E-03   0.19460573E-03   0.15023566E-05   0.13877516E-05   0.15447583E-05
0.24386680E-02   0.12355946E-02   0.30308565E-03   0.23904479E-05   0.15447594E-05   0.24242312E-05

THE TRAJECTORY HAS BEEN RECTIFIED

POSITION VECTOR (R DATE)  =    0.35286039E 04   0.26135703E 04   0.59903381E 04
VELOCITY VECTOR (V DATE)  =   -0.56663404E 01  -0.18969886E 01   0.43348668E 01

ELLIPTIC ORBIT ELEMENTS

SEMIMAJOR AXIS       (KM)  =   0.75441295E 04
ECCENTRICITY               =   0.24129352E-01
TRUE ANOMOLY DATE    (DEG) =   0.51175177E 02
R ASC OF ASC NODE    (DEG) =   0.24743707E 02
ARG OF PERIAPSE      (DEG) =   0.34623335E 01
ORBIT INCLINATION    (DEG) =   0.81486699E 02
DEL NODE PER REV     (RAD) =  -0.10806019E-02
DEL APSE PER REV     (RAD) =  -0.32497919E-02
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,   0.56945976E 05   SEC (U.T.)
R VECTOR (1950)        =   0.34884085E 04   0.25822898E 04   0.60304252E 04
V VECTOR (1950)        =  -0.56993252E 01  -0.19020114E 01   0.42825333E 01
R VECTOR (DATE)        =   0.34709745E 04   0.25938564E 04   0.60355187E 04
V VECTOR (DATE)        =  -0.56991529E 01  -0.19211317E 01   0.42742197E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.97422131E 03   0.12049063E 04   0.16918864E 04
    RELATIVE VELOCITY  =  -0.55980414E 01  -0.22261601E 01   0.42825333E 01
    R, RDOT, AZ, ELEV  =   0.22942070E 04   0.43662043E 01   0.41577024E 02   0.18875207E 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,   0.56955966E 05   SEC (U.T.)
R VECTOR (1950)        =   C.34313005E 04   0.25631628E 04   0.60729177E 04
V VECTOR (1950)        =  -0.57328205E 01  -0.19269179E 01   0.42238132E 01
R VECTOR (DATE)        =   0.34138691E 04   0.25745379E 04   0.60779279E 04
V VECTOR (DATE)        =  -0.57324793E 01  -0.19461493E 01   0.42154506E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.10303137E 04   0.11823493E 04   0.17342955E 04
    RELATIVE VELOCITY  =  -0.56313005E 01  -0.22509928E 01   0.42238132E 01
    R, RDOT, AZ, ELEV  =   0.23382209E 04   0.44760041E 01   0.40270426E 02   0.18141811E 02

UNDRFLOW AT 33550 IN MQ
```

EPOCH (REL. TO 1950.0)	=	0.55950000E 04	0.73621258E 00	(DAYS,DAYS)
EPOCH (REL. JD 2433282.5)	=	0.56955966E 05	0.55950000E 04	(DAYS U.T.)

POSITION VECTOR (DATE)	=	0.34134202E 04	0.25743134E 04	0.60778997E 04
VELOCITY VECTOR (DATE)	=	-0.57328903E 01	-0.19464139E 01	0.42150432E 01
DELTA [illegible]	=	-0.44887463E 00	-0.22444469E 00	-0.28121101E-01
DELTA VELOCITY	=	-0.41108603E-03	-0.26465696E-03	-0.40736123E-03

THE COVARIANCE MATRIX FOR THE ERRORS IN STATE ** P **

0.27705453[illegible] 01	0.12543902E 01	0.21577582E 00	0.24427052E-02	0.17479012E-02	0.24603880E-02
0.12543973[illegible] 01	0.74213018E 00	0.11028972E 00	0.12390401E-02	0.73064370E-03	0.12510759E-02
0.215780[illegible] 00	0.11028659E 00	0.17051840E 00	0.27263621E-03	0.22603810E-03	0.35674149E-03
0.244270[illegible]-02	0.12390195E-02	0.27263264E-03	0.23356424E-05	0.14477700E-05	0.23694076E-05
0.1747[illegible]-02	0.73063723E-03	0.22603354E-03	0.14477872E-05	0.13690418E-05	0.15491357E-05
0.24603806E-02	0.12510747E-02	0.35673125E-03	0.23693988E-05	0.15491370E-05	0.24861255E-05

THE TRAJECTORY HAS BEEN RECTIFIED

POSITION VECTOR (R DATE)	=	0.34134202E 04	0.25743134E 04	0.60778997E 04
VELOCITY VECTOR (V DATE)	=	-0.57328903E 01	-0.19464139E 01	0.42150432E 01

ELLIPTIC ORBIT ELEMENTS

SEMIMAJOR AXIS	(KM)	=	0.75407120E 04
ECCENTRICITY		=	0.23896959E-01
TRUE ANOMOLY DATE	(DEG)	=	0.53578872E 02
R ASC OF ASC NODE	(DEG)	=	0.24735458E 02
ARG OF PERIAPSE	(DEG)	=	0.22157463E 01
ORBIT INCLINATION	(DEG)	=	0.81486210E 02
DEL NODE PER REV	(RAD)	=	-0.10816191E-02
DEL APSE PER REV	(RAD)	=	-0.32526197E-02

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,   0.56965978E 05   SEC (U.T.)
R VECTOR (1950)        =   0.33724666E 04   0.25429144E 04    0.61169966E 04
V VECTOR (1950)        =  -0.57648872E 01  -0.19506815E 01    0.41619842E 01
R VECTOR (DATE)        =   0.33550392E 04   0.25540920E 04    0.61219210E 04
V VECTOR (DATE)        =  -0.57643763E 01  -0.19700191E 01    0.41535748E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.10881265E 04   0.11586608E 04    0.17782886E 04
    RELATIVE VELOCITY  =  -0.56631307E 01  -0.22746823E 01    0.41619842E 01
    R, RDOT, AZ, ELEV  =   0.23851256E 04   0.45816564E 01    0.38974083E 02   0.17407458E 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,   0.56975989E 05   SEC (U.T.)
R VECTOR (1950)        =   0.33145901E 04   0.25232626E 04    0.61583655E 04
V VECTOR (1950)        =  -0.57972717E 01  -0.19752160E 01    0.41024017E 01
R VECTOR (DATE)        =   0.32971686E 04   0.25342461E 04    0.61632054E 04
V VECTOR (DATE)        =  -0.57965919E 01  -0.19946609E 01    0.40939450E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.11449776E 04   0.11355731E 04    0.18195730E 04
    RELATIVE VELOCITY  =  -0.56952789E 01  -0.22991424E 01    0.41024017E 01
    R, RDOT, AZ, ELEV  =   0.24313259E 04   0.46784109E 01    0.37786582E 02   0.16683124E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0    ) =     0.55950000E 04    0.73644432E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =      0.56975989E 05    0.55950000E 04 (DAYS, U.T.)

POSITION VECTOR ( DATE )  =      0.32965969E 04    0.25339565E 04    0.61631596E 04
VELOCITY VECTOR ( DATE )  =     -0.57971071E 01   -0.19949929E 01    0.40934160E 01
DELTA RADIUS              =     -0.57171763E 00   -0.28962828E 00   -0.45720529E-01
DELTA VELOCITY            =     -0.51526346E-03   -0.33204081E-03   -0.52893053E-03


THE COVARIANCE MATRIX FOR THE ERRORS IN STATE   ** P **


0.27603657E 01   0.12586661E 01   0.26485649E 00   0.23865334E-02   0.17287414E-02   0.24955346E-02
0.12584732E 01   0.74294309E 00   0.13709713E 00   0.12170754E-02   0.73061915E-03   0.12737991E-02
0.26486095E 00   0.13709418E 00   0.18549937E 00   0.31725331E-03   0.25703402E-03   0.41219511E-03
0.23865368E-02   0.12170551E-02   0.31724988E-03   0.22448427E-05   0.14021140E-05   0.23596441E-05
0.17287435E-02   0.73061274E-03   0.25702926E-03   0.14021304E-05   0.13553901E-05   0.15609523E-05
0.24955271E-02   0.12737983E-02   0.41218471E-03   0.23596351E-05   0.15609537E-05   0.25613267E-05


THE TRAJECTORY HAS BEEN RECTIFIED

POSITION VECTOR (R DATE)  =      0.32965969E 04    0.25339565E 04    0.61631596E 04
VELOCITY VECTOR (V DATE)  =     -0.57971071E 01   -0.19949929E 01    0.40934160E 01


ELLIPTIC ORBIT ELEMENTS

SEMIMAJOR AXIS       (KM)  =     0.75369213E 04
ECCENTRICITY               =     0.23666219E-01
TRUE ANOMOLY DATE    (DEG) =     0.56106776E 02
R ASC OF ASC NODE    (DEG) =     0.24726917E 02
ARG OF PERIAPSE      (DEG) =     0.84623569E 00
ORBIT INCLINATION    (DEG) =     0.81485578E 02
DEL NODE PER REV     (RAD) =    -0.10827633E-02
DEL APSE PER REV     (RAD) =    -0.32557619E-02
```

```
TIME FROM JD 2433282.5 =   0.55950000F 04   DAYS           ,   0.56985980E 05   SEC (U.T.)
R VECTOR (1950)        =   0.32551435E 04   0.25025367E 04    0.62010805E 04
V VECTOR (1950)        =  -0.58279692E 01  -0.19983996E 01    0.40399138F 01
R VECTOR (DATE)        =   0.32377297E 04   0.25133208E 04    0.62058337E 04
V VECTOR (DATE)        =  -0.58271207E 01  -0.20179461E 01    0.40314122E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.12033961E 04   0.11114114E 04    0.18622014E 04
    RELATIVE VELOCITY  =  -0.57257403E 01  -0.23222516E 01    0.40399138E 01
    R, RDOT, AZ, ELEV  =   0.24801595E 04   0.47708583E 01    0.36611534E 02   0.15961072E 02

TIME FROM JD 2433282.5 =   0.55950000F 04   DAYS           ,   0.56995971E 05   SEC (U.T.)
R VECTOR (1950)        =   0.31967621E 04   0.24824510E 04    0.62411417E 04
V VECTOR (1950)        =  -0.58591014E 01  -0.20224530E 01    0.39797414E 01
R VECTOR (DATE)        =   0.31793576E 04   0.24930393E 04    0.62458097E 04
V VECTOR (DATE)        =  -0.58580847E 01  -0.20421027E 01    0.39711944E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.12607456E 04   0.10878943E 04    0.19021773E 04
    RELATIVE VELOCITY  =  -0.57566366E 01  -0.23462304E 01    0.39797414E 01
    R, RDOT, AZ, ELEV  =   0.25280965E 04   0.48555809E 01    0.35535404E 02   0.15252904E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0    ) =    0.55950000E 04   0.73667559E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =    0.56995971E 05   0.55950000E 04 (DAYS  U.T.)

POSITICN VECTOR ( DATE )  =    0.31787537E 04   0.24927308E 04   0.62457495E 04
VELOCITY VECTOR ( DATE )  =   -C.58586195E 01  -0.20424499E 01   0.39706247E 01
DELTA RADIUS              =   -0.60382378E 00  -0.30849953E 00  -0.60177338E-01
DELTA VELOCITY            =   -C.53482179E-03  -0.34723013E-03  -0.56965912E-03


THE CNVARIANCE MATRIX FOR THE ERRORS IN STATE  ** P **


0.27622888E 01   0.12692693E 01   0.31438887E 00   0.23413497E-02   0.17162238E-02   0.25414236E-02
0.12692763E 01   0.74733364E 00   0.16437305E 00   0.12006158E-02   0.73424805E-03   0.13024770E-02
0.31439336E 00   0.16437028E 00   0.20247935E 00   0.36056197E-03   0.28782678E-03   0.46983602E-03
0.23413535E-02   0.12005958E-02   0.36055870E-03   0.21657198E-05   0.13630696E-05   0.23582447E-05
0.17162260E-02   0.73424172E-03   0.28782184E-03   0.13630852E-05   0.13454219E-05   0.15785070E-05
0.25414161E-02   0.13024764E-02   0.46982546E-03   0.23582355E-05   0.15785085E-05   0.26478694E-05


THE TRAJECTORY HAS BEEN RECTIFIED

POSITION VECTOR (R DATE)  =    0.31787537E 04   0.24927308E 04   0.62457495E 04
VELOCITY VECTOR (V DATE)  =   -0.58586195E 01  -0.20424499E 01   0.39706247E 01


ELLIPTIC ORBIT ELEMENTS

SEMIMAJOR AXIS      (KM)  =    0.75328267E 04
ECCENTRICITY              =    0.23447670E-01
TRUE ANOMOLY DATE   (DEG) =    0.58766999F 02
R ASC OF ASC NODE   (DEG) =    0.24718090E 02
ARG OF PERIAPSE     (DEG) =    0.35934032E 03
ORBIT INCLINATION   (DEG) =    0.81484786E 02
DEL NODE PER REV    (RAD) =   -0.10840184E-02
DEL APSE PER REV    (RAD) =   -0.32591611E-02
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,   0.57005982E 05   SEC (U.T.)
R VECTOR (1950)        =   0.31365851E 04   0.24612258E 04   0.62826525E 04
V VECTOR (1950)        =  -0.58884692E 01  -0.20450950E 01   0.39164737E 01
R VECTOR (DATE)        =   0.31191917E 04   0.24716123E 04   0.62872328E 04
V VECTOR (DATE)        =  -0.58872845E 01  -0.20648419E 01   0.39078839E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.13198851E 04   0.10632276E 04   0.19436004E 04
    RELATIVE VELOCITY  =  -0.57857681E 01  -0.23687976E 01   0.39164737E 01
    R, RDOT, AZ, ELEV  =   0.25787850E 04   0.49364479E 01   0.34465827E 02   0.14545728E 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,   0.57015993E 05   SEC (U.T.)
R VECTOR (1950)        =   0.30774833E 04   0.24406331E 04   0.63215564E 04
V VECTOR (1950)        =  -0.59184675E 01  -0.20687580E 01   0.38554908E 01
R VECTOR (DATE)        =   0.30601025E 04   0.24508214E 04   0.63260504E 04
V VECTOR (DATE)        =  -0.59171148E 01  -0.20886042E 01   0.38468571E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.13779439E 04   0.10391932E 04   0.19824180E 04
    RELATIVE VELOCITY  =  -0.58155299E 01  -0.23923854E 01   0.38554908E 01
    R, RDOT, AZ, ELEV  =   0.26284279E 04   0.50107966E 01   0.33485115E 02   0.13853421E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0    ) =    0.55950000E 04   0.73690733E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =    0.57015993E 05   0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )  =    0.30595448E 04   0.24505349E 04   0.63259828E 04
VELOCITY VECTOR ( DATE )  =   -0.59175997E 01  -0.20889229E 01   0.38463205E 01
DELTA RADIUS              =   -0.55769007E 00  -0.28648026E 00  -0.67545458E-01
DELTA VELOCITY            =   -0.48489820E-03  -0.31875257E-03  -0.53660038E-03
```

THE COVARIANCE MATRIX FOR THE ERRORS IN STATE ** P **

0.27743038E 01	0.12852868E 01	0.36511265E 00	0.23045737E-02	0.17089784E-02	0.25969593E-02
0.12852937E 01	0.75488210E 00	0.19252277E 00	0.11884237E-02	0.74090135E-03	0.13366604E-02
0.36511715E 00	0.19252019E 00	0.22161837E 00	0.40315890E-03	0.31882355E-03	0.53044844E-03
0.23045779E-02	0.11884039E-02	0.40315582E-03	0.20952898E-05	0.13289720E-05	0.23635036E-05
0.17089807E-02	0.74089507E-03	0.31881843E-03	0.13289869E-05	0.13382444E-05	0.16009703E-05
0.25969518E-02	0.13366600E-02	0.53043775E-03	0.23634944E-05	0.16009719E-05	0.27454739E-05

THE TRAJECTORY HAS BEEN RECTIFIED

```
POSITION VECTOR (R DATE)  =    0.30595448E 04   0.24505349E 04   0.63259828E 04
VELOCITY VECTOR (V DATE)  =   -0.59175997E 01  -0.20889229E 01   0.38463205E 01
```

ELLIPTIC ORBIT ELEMENTS

```
SEMIMAJOR AXIS        (KM)  =    0.75284863E 04
ECCENTRICITY                =    0.23252392E-01
TRUE ANOMOLY DATE     (DEG) =    0.61574044E 02
R ASC OF ASC NODE     (DEG) =    0.24708990E 02
ARG OF PERIAPSE       (DEG) =    0.35768756E 03
ORBIT INCLINATION     (DEG) =    0.81483818E 02
DEL NODE PER REV      (RAD) =   -0.10853715E-02
DEL APSE PER REV      (RAD) =   -0.32627700E-02
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS         ,   0.57025984E 05   SEC (U.T.)
R VECTOR (1950)        =   0.30169268E 04   0.24190469E 04   0.63616956E 04
V VECTOR (1950)        =  -0.59463023E 01  -0.20907060E 01   0.37918120E 01
R VECTOR (DATE)        =   0.29995603E 04   0.24290321E 04   0.63661013E 04
V VECTOR (DATE)        =  -0.59447833E 01  -0.21106443E 01   0.37831378E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.14374562E 04   0.10141705E 04   0.20224689E 04
    RELATIVE VELOCITY  =  -0.58431289E 01  -0.24142584E 01   0.37918120E 01
    R, RDOT, AZ, ELEV  =   0.26805228E 04   0.50809469E 01   0.32514727E 02   0.13167846E 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS         ,   0.57035975E 05   SEC (U.T.)
R VECTOR (1950)        =   0.29573751E 04   0.23980434E 04   0.63992714E 04
V VECTOR (1950)        =  -0.59750364E 01  -0.21138736E 01   0.37302997E 01
R VECTOR (DATE)        =   0.29400246E 04   0.24078289E 04   0.64035902E 04
V VECTOR (DATE)        =  -0.59733502E 01  -0.21339070E 01   0.37215835E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.14959614E 04   0.98973959E 03   0.20599578E 04
    RELATIVE VELOCITY  =  -0.58716276E 01  -0.24373508E 01   0.37302997E 01
    R, RDOT, AZ, ELEV  =   0.27314668E 04   0.51458232E 01   0.31622941E 02   0.12495597E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0     ) =    0.55950000E 04   0.73713860E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =    0.57035975E 05   0.55950000E 04 (DAYS  U.T.)

POSITICN VECTOR ( DATE )  =    0.29394812E 04   0.24075481E 04   0.64035130E 04
VELOCITY VECTOR ( DATE )  =   -0.59738138E 01  -0.21342152E 01   0.37210506E 01
DELTA RADIUS              =   -0.54335234E 00  -0.28086163E 00  -0.77169575E-01
DELTA VELOCITY            =   -0.46364703E-03  -0.30827842E-03  -0.53281809E-03
```

THE COVARIANCE MATRIX FOR THE ERRORS IN STATE ** P **

0.27941307E 01	0.13055612E 01	0.41716483E 00	0.22739150E-02	0.17056391E-02	0.26602697E-02
0.13055681E 01	0.76503716E 00	0.22162971E 00	0.11793265E-02	0.74985150E-03	0.13754372E-02
0.41716933E 00	0.22162733E 00	0.24294825E 00	0.44508622E-03	0.35006001E-03	0.59418154E-03
0.22739196E-02	0.11793071E-02	0.44508334E-03	0.20313035E-05	0.12984798E-05	0.23734746E-05
0.17056415E-02	0.74984529E-03	0.35005473E-03	0.12984939E-05	0.13330193E-05	0.16271736E-05
0.26602623E-02	0.13754372E-02	0.59417071E-03	0.23734652E-05	0.16271753E-05	0.28526530E-05

THE TRAJECTORY HAS BEEN RECTIFIED

```
POSITICN VECTOR (R DATE)  =    0.29394812E 04   0.24075481E 04   0.64035130E 04
VELOCITY VECTOR (V DATE)  =   -0.59738138E 01  -0.21342152E 01   0.37210506E 01
```

ELLIPTIC ORBIT ELEMENTS

```
SEMIMAJOR AXIS        (KM)  =    0.75238893E 04
ECCENTRICITY                =    0.23084203E-01
TRUE ANOMOLY DATE     (DEG) =    0.64516151E 02
R ASC OF ASC NODE     (DEG) =    0.24699603E 02
ARG OF PERIAPSE       (DEG) =    0.35589520E 03
ORBIT INCLINATION     (DEG) =    0.81482684E 02
DEL NODE PER REV      (RAD) =   -C.10868249E-02
DEL APSE PER REV      (RAD) =   -0.32666009E-02
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS           ,   0.57045986E 05   SEC (U.T.)
R VECTOR (1950)        =   0.28961942E 04   0.23760089E 04     0.64381857E 04
V VECTOR (1950)        =  -0.60014846E 01  -0.21352338E 01     0.36659744E 01
R VECTOR (DATE)        =   0.28788613E 04   0.23855893E 04     0.64424153E 04
V VECTOR (DATE)        =  -0.59996333E 01  -0.21553545E 01     0.36572196E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.15560873E 04   0.96425844E 03     0.20987829E 04
    RELATIVE VELOCITY  =  -0.58978395E 01  -0.24586353E 01     0.36659744E 01
    R, RDOT, AZ, ELEV  =   0.27849760E 04   0.52068231E 01     0.30736872E 02   0.11829161E 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS           ,   0.57055997E 05   SEC (U.T.)
R VECTOR (1950)        =   0.28359731E 04   0.23545183E 04     0.64745754E 04
V VECTOR (1950)        =  -0.60290644E 01  -0.21579935E 01     0.36037077E 01
R VECTOR (DATE)        =   0.28186595E 04   0.23638969E 04     0.64787171E 04
V VECTOR (DATE)        =  -0.60270462E 01  -0.21782055E 01     0.35949126E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.16152508E 04   0.93933028E 03     0.21350847E 04
    RELATIVE VELOCITY  =  -0.59251834E 01  -0.24813193E 01     0.36037077E 01
    R, RDOT, AZ, ELEV  =   0.28372457E 04   0.52635906E 01     0.29920804E 02   0.11174927E 02

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0     ) =    0.55950000E 04   0.73737034E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =    0.57055997E 05   0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )  =    0.28180265E 04   0.23635661E 04   0.64786158E 04
VELOCITY VECTOR ( DATE )  =   -0.60275764E 01  -0.21785592E 01   0.35942810E 01
DELTA RADIUS              =   -0.63294710E 00  -0.33075641E 00  -0.10119842E 00
DELTA VELOCITY            =   -0.53029877E-03  -0.35373760E-03  -0.63149191E-03


THE COVARIANCE MATRIX FOR THE ERRORS IN STATE  ** P **


0.28206825E 01   0.13296202E 01   0.47109100E 00   0.22478400E-02   0.17054219E-02   0.27308683E-02
0.13296272E 01   0.77759086E 00   0.25200912E 00   0.11725832E-02   0.76074922E-03   0.14186502E-02
0.47109547E 00   0.25200695E 00   0.26669639E 00   0.48669498E-03   0.38181201E-03   0.66170630E-03
0.22478449E-02   0.11725639E-02   0.48669233E-03   0.19719866E-05   0.12706005E-05   0.23871272E-05
0.17054242E-02   0.76074305E-03   0.38180656E-03   0.12706140E-05   0.13292280E-05   0.16566702E-05
0.27308609E-02   0.14186504E-02   0.66169535E-03   0.23871177E-05   0.16566720E-05   0.29695366E-05


THE TRAJECTORY HAS BEEN RECTIFIED

POSITION VECTOR (R DATE)  =    0.28180265E 04   0.23635661E 04   0.64786158E 04
VELOCITY VECTOR (V DATE)  =   -0.60275764E 01  -0.21785592E 01   0.35942810E 01


ELLIPTIC ORBIT ELEMENTS

SEMIMAJOR AXIS        (KM)  =    0.75189818E 04
ECCENTRICITY                =    0.22947108E-01
TRUE ANOMOLY DATE     (DEG) =    0.67587916E 02
R ASC OF ASC NODE     (DEG) =    0.24689898E 02
ARG OF PERIAPSE       (DEG) =    0.35397396E 03
ORBIT INCLINATION     (DEG) =    0.81481415E 02
DEL NODE PER REV      (RAD) =   -0.10883912E-02
DEL APSE PER REV      (RAD) =   -0.32707060E-02
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,    0.57065988E 05   SEC (U.T.)
R VECTOR (1950)        =   0.27743398E 04   0.23320814E 04    0.65120873E 04
V VECTOR (1950)        =  -0.60540934E 01  -0.21787177E 01    0.35389320E 01
R VECTOR (DATE)        =   0.27570471E 04   0.23412533E 04    0.65161391E 04
V VECTOR (DATE)        =  -0.60519114E 01  -0.21990123E 01    0.35301003E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.16758238E 04   0.91345622E 03    0.21725067E 04
    RELATIVE VELOCITY  =  -0.59499767E 01  -0.25019677E 01    0.35389320E 01
    R, RDOT, AZ, ELEV  =   0.28918113E 04   0.53164006E 01    0.29112322E 02   0.10528636E 02

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,    0.57075979E 05   SEC (U.T.)
R VECTOR (1950)        =   0.27137234E 04   0.23102031E 04    0.65471306E 04
V VECTOR (1950)        =  -0.60803999E 01  -0.22009692E 01    0.34761954E 01
R VECTOR (DATE)        =   0.26964533E 04   0.23191718E 04    0.65510939E 04
V VECTOR (DATE)        =  -0.60780520E 01  -0.22213507E 01    0.34673253E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.17353762E 04   0.88814537E 03    0.22074615E 04
    RELATIVE VELOCITY  =  -0.59760478E 01  -0.25241432E 01    0.34761954E 01
    R, RDOT, AZ, ELEV  =   0.29450329E 04   0.53657981E 01    0.28366888E 02   0.98951340E 01

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0    ) =   0.55950000E 04   0.73760161E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =    0.57075979E 05   0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )  =    0.26958279E 04   0.23188438E 04   0.65509799E 04
VELOCITY VECTOR ( DATE )  =   -0.60785654E 01  -0.22216987E 01   0.34666896E 01
DELTA RADIUS              =   -0.62534335E 00  -0.32805193E 00  -0.11399805E 00
DELTA VELOCITY            =   -0.51340857E-03  -0.34802775E-03  -0.63567116E-03
```

THE COVARIANCE MATRIX FOR THE ERRORS IN STATE ** P **

0.28523985E 01	0.13566470E 01	0.52685765E 00	0.22248905E-02	0.17074236E-02	0.28073461E-02
0.13566539E 01	0.79214962E 00	0.28365953E 00	0.11674189E-02	0.77308425E-03	0.14655994E-02
0.52686207E 00	0.28365757E 00	0.29290930E 00	0.52784925E-03	0.41400997E-03	0.73303183E-03
0.22248958E-02	0.11673999E-02	0.52784684E-03	0.19160357E-05	0.12445227E-05	0.24030724E-05
0.17074260E-02	0.77307814E-03	0.41400436E-03	0.12445355E-05	0.13263307E-05	0.16885930E-05
0.28073385E-02	0.14655999E-02	0.73302075E-03	0.24030628E-05	0.16885950E-05	0.30949021E-05

THE TRAJECTORY HAS BEEN RECTIFIED

```
POSITION VECTOR (R DATE)  =    0.26958279E 04   0.23188438E 04   0.65509799E 04
VELOCITY VECTOR (V DATE)  =   -0.60785654E 01  -0.22216987E 01   0.34666896E 01
```

ELLIPTIC ORBIT ELEMENTS

```
SEMIMAJOR AXIS       (KM)  =    0.75138331E 04
ECCENTRICITY               =    0.22849821E-01
TRUE ANOMOLY DATE   (DEG)  =    0.70782543E 02
R ASC OF ASC NODE   (DEG)  =    0.24679902E 02
ARG OF PERIAPSE     (DEG)  =    0.35192522E 03
ORBIT INCLINATION   (DEG)  =    0.81479989E 02
DEL NODE PER REV    (RAD)  =   -0.10900547E-02
DEL APSE PER REV    (RAD)  =   -0.32750264E-02
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS           ,    0.57085969E 05   SEC (U.T.)
R VECTOR (1950)        =   0.26516435E 04   0.22873874E 04     0.65833217E 04
V VECTOR (1950)        =  -0.61039841E 01  -0.22210473E 01     0.34109822E 01
R VECTOR (DATE)        =   0.26343975E 04   0.22961480E 04     0.65871945E 04
V VECTOR (DATE)        =  -0.61014740E 01  -0.22415067E 01     0.34020776E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.17963881E 04   0.86189261E 03     0.22435621E 04
    RELATIVE VELOCITY  =  -0.59993965E 01  -0.25441453E 01     0.34109822E 01
    R, RDOT, AZ, ELEV  =   0.30005733E 04   0.54113708E 01     0.27626917E 02   0.92694284E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS           ,    0.57095960E 05   SEC (U.T.)
R VECTOR (1950)        =   0.25905346E 04   0.22650886E 04     0.66170837E 04
V VECTOR (1950)        =  -0.61290687E 01  -0.22428306E 01     0.33476773E 01
R VECTOR (DATE)        =   0.25733145E 04   0.22736444E 04     0.66208674E 04
V VECTOR (DATE)        =  -0.61263936E 01  -0.22633729E 01     0.33387361E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.18564248E 04   0.83616104E 03     0.22772350E 04
    RELATIVE VELOCITY  =  -0.60242457E 01  -0.25658523E 01     0.33476773E 01
    R, RDOT, AZ, ELEV  =   0.30547140E 04   0.54543703E 01     0.26942994E 02   0.86549325E 01

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0      ) =    0.55950000E 04    0.73783287E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =    0.57095960E 05    0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )  =    0.25728327E 04    0.22733910E 04    0.66207682E 04
VELOCITY VECTOR ( DATE )  =   -0.61267809E 01   -0.22636406E 01    0.33382370E 01
DELTA RADIUS              =   -0.48185023E 00   -0.25341403E 00   -0.99179737E-01
DELTA VELOCITY            =   -0.38733633E-03   -0.26774046E-03   -0.49910284E-03


THE COVARIANCE MATRIX FOR THE ERRORS IN STATE  ** P **


0.28885162E 01   0.13862959E 01   0.58477246E 00   0.22040578E-02   0.17111127E-02   0.28892790E-02
0.13863029E 01   0.80856143E 00   0.31677339E 00   0.11633261E-02   0.78660216E-03   0.15161323E-02
0.58477682E 00   0.31677166E 00   0.32179832E 00   0.56866270E-03   0.44678042E-03   0.80860316E-03
0.22040633E-02   0.11633073E-02   0.56866053E-03   0.18623793E-05   0.12196247E-05   0.24205223E-05
0.17111152E-02   0.78659612E-03   0.44677467E-03   0.12196369E-05   0.13239851E-05   0.17225761E-05
0.28892714E-02   0.15161331E-02   0.80859198E-03   0.24205127E-05   0.17225783E-05   0.32287317E-05


THF TRAJECTORY HAS BEEN RECTIFIED

POSITION VECTOR (R DATE)  =    0.25728327E 04    0.22733910E 04    0.66207682E 04
VELOCITY VECTOR (V DATE)  =   -0.61267809E 01   -0.22636406E 01    0.33382370E 01


ELLIPTIC ORBIT ELEMENTS

SEMIMAJOR AXIS       (KM)  =    0.75085379E 04
ECCENTRICITY               =    0.22808863E-01
TRUE ANOMOLY DATE    (DEG) =    0.74103637E 02
R ASC OF ASC NODE    (DEG) =    0.24669619E 02
ARG OF PERIAPSE      (DEG) =    0.34974668E 03
ORBIT INCLINATION    (DEG) =    0.81478375E 02
DEL NODE PER REV     (RAD) =   -0.10917938E-02
DEL APSE PER REV     (RAD) =   -0.32794831E-02
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,    0.57105971E 05   SEC (U.T.)
R VECTOR (1950)        =   0.25280493E 04   0.22419344E 04    0.66520438E 04
V VECTOR (1950)        =  -0.61511484E 01  -0.22622258E 01    0.32820832E 01
R VECTOR (DATE)        =   0.25108563E 04   0.22502808E 04    0.66557363E 04
V VECTOR (DATE)        =  -0.61483129E 01  -0.22828409E 01    0.32731098E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.19178329E 04   0.80956525E 03    0.23121039E 04
    RELATIVE VELOCITY  =  -0.60460898E 01  -0.25851709E 01    0.32820832E 01
    R, RDOT, AZ, ELEV  =   0.31111579E 04   0.54934669E 01    0.26264058E 02   0.80506172E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,    0.57115982E 05   SEC (U.T.)
R VECTOR (1950)        =   0.24663482E 04   0.22191795E 04    0.66845818E 04
V VECTOR (1950)        =  -0.61750538E 01  -0.22835775E 01    0.32181089E 01
R VECTOR (DATE)        =   0.24491844E 04   0.22273191E 04    0.66881843E 04
V VECTOR (DATE)        =  -0.61720536E 01  -0.23042715E 01    0.32091006E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.19784524E 04   0.78337245E 03    0.23445519E 04
    RELATIVE VELOCITY  =  -0.60697595E 01  -0.26064458E 01    0.32181089E 01
    R, RDOT, AZ, ELEV  =   0.31662075E 04   0.55308869E 01    0.25633863E 02   0.74532810E 01

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0    ) =    0.55950000E 04    0.73806462E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =    0.57115982E 05    0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )  =    0.24484998E 04    0.22269554E 04    0.66880309E 04
VELOCITY VECTOR ( DATE )  =   -0.61725928E 01   -0.23046460E 01    0.32083800E 01
DELTA RADIUS              =   -0.68452439E 00   -0.36373093E 00   -0.15333930E 00
DELTA VELOCITY            =   -0.53930064E-03   -0.37457054E-03   -0.72058519E-03


THE COVARIANCE MATRIX FOR THE ERRORS IN STATE  ** P **


0.29284455E 01   0.14183081E 01   0.64512777E 00   0.21845135E-02   0.17160612E-02   0.29763720E-02
0.14183152E 01   0.82672034E 00   0.35154020E 00   0.11598835E-02   0.80110061E-03   0.15701654E-02
0.64513206E 00   0.35153871E 00   0.35361003E 00   0.60920359E-03   0.48023167E-03   0.88887890E-03
0.21845193E-02   0.11598649E-02   0.60920168E-03   0.18101612E-05   0.11954026E-05   0.24388036E-05
0.17160638E-02   0.80109463E-03   0.48022576E-03   0.11954141E-05   0.13219129E-05   0.17583256E-05
0.29763644E-02   0.15701664E-02   0.88886763E-03   0.24387940E-05   0.17583279E-05   0.33711237E-05


THE TRAJECTORY HAS BEEN RECTIFIED

POSITION VECTOR (R DATE)  =    0.24484998E 04    0.22269554E 04    0.66880309E 04
VELOCITY VECTOR (V DATE)  =   -0.61725928E 01   -0.23046460E 01    0.32083800E 01


ELLIPTIC ORBIT ELEMENTS

SEMIMAJOR AXIS       (KM)  =    0.75028958E 04
ECCENTRICITY               =    0.22813109E-01
TRUE ANOMOLY DATE    (DEG) =    0.77514470E 02
R ASC OF ASC NODE    (DEG) =    0.24658999E 02
ARG OF PERIAPSE      (DEG) =    0.34747976E 03
ORBIT INCLINATION    (DEG) =    0.81476669E 02
DEL NODE PER REV     (RAD) =   -0.10936542E-02
DEL APSE PER REV     (RAD) =   -0.32842571E-02
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS        ,    0.57125973E 05   SEC (U.T.)
R VECTOR (1950)        =   0.24033921E 04   0.21956092E 04   0.67180969E 04
V VECTOR (1950)        =  -0.61958063E 01  -0.23023804E 01   0.31520803E 01
R VECTOR (DATE)        =   0.23862588E 04   0.22035378E 04   0.67216075E 04
V VECTOR (DATE)        =  -0.61926469E 01  -0.23231426E 01   0.31430416E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.20403247E 04   0.75636525E 03   0.23779752E 04
    RELATIVE VELOCITY  =  -0.60902768E 01  -0.26251719E 01   0.31520803E 01
    R, RDOT, AZ, ELEV  =   0.32233180E 04   0.55644903E 01   0.25009328E 02   0.68656973E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS        ,    0.57135964E 05   SEC (U.T.)
R VECTOR (1950)        =   0.23413782E 04   0.21725026E 04   0.67492671E 04
V VECTOR (1950)        =  -0.62184300E 01  -0.23232068E 01   0.30877265E 01
R VECTOR (DATE)        =   0.23242773E 04   0.21802234E 04   0.67526873F 04
V VECTOR (DATE)        =  -0.62151073E 01  -0.23440437E 01   0.30786548E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.21012506E 04   0.72982600E 03   0.24090549E 04
    RELATIVE VELOCITY  =  -0.61126654E 01  -0.26459214E 01   0.30877265E 01
    R, RDOT, AZ, ELEV  =   0.32789397E 04   0.55968342E 01   0.24429651E 02   0.62872173E 01

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0    ) =   0.55950000E 04   0.73829588E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =   0.57135964E 05   0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )  =   0.23236313E 04   0.21798783E 04   0.67525285E 04
VELOCITY VECTOR ( DATE )  =  -0.62156053E 01  -0.23443948E 01   0.30779629E 01
DELTA RADIUS              =  -0.64597639E 00  -0.34512861E 00  -0.15871855E 00
DELTA VELOCITY            =  -0.49801426E-03  -0.35106403E-03  -0.69188405E-03


THE COVARIANCE MATRIX FOR THE ERRORS IN STATE  ** P **


0.29711736E 01   0.14521262E 01   0.70774771E 00   0.21654515E-02   0.17217255E-02   0.30675610E-02
0.14521333E 01   0.84635054E 00   0.38788556E 00   0.11566306E-02   0.81623724E-03   0.16271526E-02
0.70775190E 00   0.38788431E 00   0.38838719E 00   0.64920136E-03   0.51420925E-03   0.97373145E-03
0.21654574E-02   0.11566122E-02   0.64919972E-03   0.17588192E-05   0.11714614E-05   0.24569732E-05
0.17217281E-02   0.81623130E-03   0.51420320E-03   0.11714722E-05   0.13198104E-05   0.17952075E-05
0.30675534E-02   0.16271538E-02   0.97372007E-03   0.24569636E-05   0.17952099E-05   0.35210217E-0[illegible]


THE TRAJECTORY HAS BEEN RECTIFIED

POSITION VECTOR (R DATE)  =   0.23236313E 04   0.21798783E 04   0.67525285E 04
VELOCITY VECTOR (V DATE)  =  -0.62156053E 01  -0.23443948E 01   0.30779629E 01


ELLIPTIC ORBIT ELEMENTS

SEMIMAJOR AXIS      (KM)  =   0.74970662E 04
ECCENTRICITY              =   0.22882242E-01
TRUE ANOMOLY DATE   (DEG) =   0.81009858E 02
R ASC OF ASC NODE   (DEG) =   0.24648064E 02
ARG OF PERIAPSE     (DEG) =   0.34512325E 03
ORBIT INCLINATION   (DEG) =   0.81474813E 02
DEL NODE PER REV    (RAD) =  -0.10955995E-02
DEL APSE PER REV    (RAD) =  -0.32892124E-02
```

```
TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,   0.57142968E 05   SEC (U.T.)
R VECTOR (1950)        =   0.22967199E 04   0.21555977E 04   0.67723979E 04
V VECTOR (1950)        =  -0.62312039E 01  -0.23352007E 01   0.30407647E 01
R VECTOR (DATE)        =   0.22796422E 04   0.21631689E 04   0.67757530E 04
V VECTOR (DATE)        =  -0.62277727E 01  -0.23560795E 01   0.30316744E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.21451433E 04   0.71050868E 03   0.24321206E 04
    RELATIVE VELOCITY  =  -0.61252742E 01  -0.26578611E 01   0.30407647E 01
    R, RDOT, AZ, ELEV  =   0.33198905E 04   0.56166503E 01   0.24018591E 02   0.58882439E 01

TIME FROM JD 2433282.5 =   0.55950000E 04   DAYS          ,   0.57149971E 05   SEC (U.T.)
R VECTOR (1950)        =   0.22530245E 04   0.21391923E 04   0.67935360E 04
V VECTOR (1950)        =  -0.62463892E 01  -0.23495340E 01   0.29953891E 01
R VECTOR (DATE)        =   0.22359712E 04   0.21466170E 04   0.67968273E 04
V VECTOR (DATE)        =  -0.62428439E 01  -0.23704629E 01   0.29862766E 01

  FLOYD  OBSERVES
    RELATIVE POSITION  =  -0.21880726E 04   0.69169910E 03   0.24531949E 04
    RELATIVE VELOCITY  =  -0.61402949E 01  -0.26721404E 01   0.29953891E 01
    R, RDOT, AZ, ELEV  =   0.33592074E 04   0.56368556E 01   0.23636717E 02   0.54919364E 01

UNDRFLOW AT 33550 IN MQ
```

```
EPOCH (REL. TO 1950.0    ) =    0.55950000E 04    0.73845801E 00 (DAYS,DAYS )
EPOCH (REL. JD 2433282.5) =    0.57149971E 05    0.55950000E 04 (DAYS  U.T.)

POSITION VECTOR ( DATE )  =    0.22466714E 04    0.21524071E 04    0.67995461E 04
VELOCITY VECTOR ( DATE )  =   -0.62347068E 01   -0.23647784E 01    0.29978392E 01
DELTA RADIUS              =    0.10700156E 02    0.57901146E 01    0.27189248E 01
DELTA VELOCITY            =    0.81370109E-02    0.56845307E-02    0.11562602E-01


THE COVARIANCE MATRIX FOR THE ERRORS IN STATE  ** P **


0.29898452E 01   0.14699593E 01   0.74985681E 00   0.21423846E-02   0.17191314E-02   0.31200541E-02
0.14699663E 01   0.85725682E 00   0.41263542E 00   0.11490863E-02   0.82347584E-03   0.16614248E-02
0.74986093E 00   0.41263438E 00   0.41383446E 00   0.67443966E-03   0.53662636E-03   0.10326324E-02
0.21423908E-02   0.11490682E-02   0.67443820E-03   0.17156949E-05   0.11494708E-05   0.24588997E-05
0.17191339E-02   0.82346997E-03   0.53662020E-03   0.11494812E-05   0.13145480E-05   0.18142320E-05
0.31200464E-02   0.16614263E-02   0.10326210E-02   0.24588899E-05   0.18142345E-05   0.36157556E-05


THE TRAJECTORY HAS BEEN RECTIFIED

POSITION VECTOR (R DATE)  =    0.22466714E 04    0.21524071E 04    0.67995461E 04
VELOCITY VECTOR (V DATE)  =   -0.62347068E 01   -0.23647784E 01    0.29978392E 01


ELLIPTIC ORBIT ELEMENTS

SEMIMAJOR AXIS       (KM)  =    0.74979162E 04
ECCENTRICITY               =    0.23690920E-01
TRUE ANOMOLY DATE    (DEG) =    0.84776797E 02
R ASC OF ASC NODE    (DEG) =    0.24637861E 02
ARG OF PERIAPSE      (DEG) =    0.34207569E 03
ORBIT INCLINATION    (DEG) =    0.81469845E 02
DEL NODE PER REV     (RAD) =   -0.10960673E-02
DEL APSE PER REV     (RAD) =   -0.32882445E-02
```

5.0 CONCLUSIONS AND RECOMMENDATIONS

The program described in the previous pages is an accurate and efficient tool for estimating the trajectory of a satellite traveling about the earth. There are, however, several features which should be incorporated in the program to extend the number of applications and to further enhance the accuracy. The most important of these are itemized below (not in the order of importance).

1) The present program prints at intervals determined by the step size employed for the numerical integration. This mode of operation is completely adequate for applications of the type of the sample problem. However, should the program be applied to general trajectory problems (a simple task implemented by replacing the present subroutine DATAPE with a dummy which reads an end time and the subroutine FILTER with a routine which terminates operation if called), prints at regular prescribed intervals would normally be required, thus, an alternate logic for this application is recommended.

2) The present logic employed in the integration cycle involves restarting the integration with the most recent values of $\vec{r}$ and $\vec{v}$ at those times when the step size has been changed. An improved logic would be to use the updated differences and sums for an epoch 3 steps before the most recent point to reevaluate the $\vec{r}$ and $\vec{v}$ at the time in the past using the central difference formula for integration. This procedure would assure that errors introduced in the extrapolation of $\vec{r}$ and $\vec{v}$ could be controlled to a higher degree.

3) The present starter routine for the numerical integration is a 4th order Runge-Kutta. This routine provides a series of values of $\vec{r}$, $\vec{v}$ and $\vec{a}$ which are differenced and summed to construct the table required in the Gauss-Jackson stepping. However, the Gauss routine operating with the predictor cycle done is a more precise routine than the R-K starter. Thus, if extreme precision is desired, the difference table provided by the start cycle must be corrected. This correction could be achieved by assuming the 6th difference is constant, constructing the missing differences back to $\vec{r}_o$, $\vec{v}_o$, and employing the central difference formula for reestimating the values of $\vec{r}$ and $\vec{v}$ to be used for the difference table.

4) The constants of the math model (the coefficients of the potential function, the gravitational constant, the quantity C_DA/M, etc.) could be estimated at the same time the state is being differentially corrected to adjust for slight bias errors. This mode of operation would assure that the trajectory could be estimated to a higher degree of precision than presently obtainable. Alternate logic would, of course, be required to determine when sufficient information was available in the data to make such a process convergent.

5) The present logic makes no attempt to correct the raw data for any known biases, for biases of the type displayed in Figures 2 and 5 (discussed in the sample problem), for time of signal propagation, for refraction, etc. Thus, these effects must be reflected in the data prior to processing in the PREPROCESSOR. Slight modification of the preliminary logic could, however, effect this correction and save considerable effort at the recording station. Thus, if corrections to the data are worthy of inclusion, these changes must be made.

6) Some editing of the raw data is presently accomplished (the observations are made "continuous" functions of time; e.g., if the data are increasing toward 2π and a value of slightly more than zero is encountered, adjustment will be made to assure that all data fall near some continuous curve before smoothing). However, no editing to determine if time is recorded properly is attempted. Several tests of this latter nature should be devised to discard the bad data points so that the possibility of effecting the smoothed data files can be disregarded (see the discussion of the preliminary processor in the sample problem).

7) The present system of providing the input data might conceivably be revised to simplify operation. Particular attention should be directed to the manner in which time is input. This data could be provided in a more straightforward manner in the form of year, month, day, hour, second, or Julian date.

Other modifications would of course be required to allow lunar and planetary trajectories to be analyzed. These modifications would all stem from the inclusion of an accurate ephemeris routine and the logic necessary to produce information in other coordinate systems. Should this task be adjudged worthy of attention, simultaneous effort should be directed toward the inclusion of midcourse guidance logic so that simulations of true trajectories and studies of trajectory shaping can be effected with efficiency.

The application of this program to real problems (such as that which was attempted in the sample) requires precise data for the station, for the station errors, for the noise, and for the observations (this latter point was discussed in items 5 and 6). However, no means is provided within the logic as a check on these data. Thus, it is considered essential that the Facility be periodically calibrated against the remainder of the tracking network, and that the results of this calibration be utilized in preference to assumed data of the type built into subroutine INPUT.

Finally, it is recommended that another sample problem be prepared. This new sample should utilize precisely reduced observed data (of all six types permissible) from several tracking stations (acquired over a period of several orbits), precise station data, precise noise data, and precise error data for the station's position. Only in this manner can a final check be performed.

APPENDIX I

CØMMØN

CØMMØN data storage is affected through the facility of labeled (non-blank) and unlabeled(blank) CØMMØN, A detailed description of the contents of both storage facilities is contained within this section.

Blank CØMMØN Map.

A blank (unlabeled)CØMMØN section is utilized as the principal means of communication between the various subprograms. This section, specified by the 525 element DATA array, is composed of the following five subarrays.

CØN: A 15 element subarray, loaded at DATA (1), containing program constants and conversion factors.

SAT: A 20 element subarray, loaded at DATA (16), containing data pertaining to the satellite at the initial epoch.

SDA: A 250 element subarray, loaded at DATA (36), containing data pertaining to the tracking stations.

STT. A 105 element subarray, loaded at DATA (286), containing statistical information and related data for the satellite.

WRK: A 135 element subarray loaded at DATA (391). The primary function of the array is to act as a "scratch paper" or working array for communication between the various subroutines.

The following map, which describes the blank CØMMØN region in detail, utilizes standard FORTRAN nomenclature with a single exception. When a description refers to a series of locations in a DATA subarray (usually time), an abbreviated notation is used, in that, reference to the two cells WRK (50) and WRK (51) is written WRK (50,51). From the context, it will be apparent that WRK (50,51) is not a single element of a double subscripted array.

Blank CØMMØN Map

DATA Location	Array Subarray	FØRTRAN Names	Description
1	CØN (1)	RE	Equatorial radii for reference ellipsoid. (Km)
2	CØN (2)	RPØL	Polar radii for reference ellipsoid. (Km)
3	CØN (3)	CØEFJ, CJ	Second coefficient of Jeffrey's potential function.
4	CØN (4)	CØEFH, CH	Third coefficient of Jeffrey's potential function.
5	CØN (5)	CØEFD, CD	Fourth coefficient of Jeffrey's potential function.
6	CØN (6)	GMERTH, GM, GME	Earth gravitational constant (Km^3/sec^2)
7	CØN (7)	ØMEGA	Earth spin rate (rad/sec)
8	CØN (8)	GMMØØN, UM, GMM	Lunar gravitational constant (Km^3/sec^2)
9	CØN (9)	GMSUN, US, GMS	Solar gravitational constant (Km^3/sec^2)
10	CØN (10)	AU, REF	Astronomical unit. (Km)
11	CØN (11)	CØNDAY, CØNV1	Conversion factor, mean solar days to sec.
12	CØN (12)	CØNV2	Conversion factor, degrees to radians
13	CØN (13)	NIN	Logical system input tape drive unit
14	CØN (14)	NØUT	Logical system output tape drive unit
15	CØN (15)		Not used
16	SAT (1)	SMASS	Satellite mass (Kg)
17	SAT (2)	AREA, A1	Cross sectional area of spherical satellite (M^2)
18	SAT (3)	CD	Satellite drag coefficient
19	SAT (4)	REFLEK, R1	Satellite surface reflectivity
20	SAT (5)	RVEC (1), R (1)	Position and velocity vectors (or orbital elements) in the true equator of date frame at the initial epoch. See SAT (13), (Km-sec degree units)
21	SAT (6)	RVEC (2), R (2)	
22	SAT (7)	RVEC (3), R (3)	
23	SAT (8)	VVEC (1), V (1)	
24	SAT (9)	VVEC (2), V (2)	
25	SAT (10)	VVEC (3), V (3)	
26	SAT (11)	DJ, TXW	Whole day } Time referenced to J. D. 2433282.423 (day).
27	SAT (12)	DJF, TXF	Fractional day }
28	SAT (13)	WINDEX	Indicator defining SAT (5) - SAT (10). 1, cartesian vectors. 2, Orbital elements.
29	SAT (14)	CHECK	Indicates whether each station is to be checked at each integration step. 1, yes. 2, no.

DATA Location	Array Subarray	FØRTRAN Names	Description	
30	SAT (15)	GØNØ	Indicates whether all stations are to be checked at the time data is available. 1, yes. 2, no.	
31	SAT (16)	CODUMP	Control option used to dump the initial DATA array (Blank CØMMØN); 0 = no dump; non zero = dump	
32	SAT (17)	KCHECK	Fixed point equivalent of SAT (14)	
33	SAT (18)	NØGØ	Fixed point equivalent of SAT (15)	
34	SAT (19)		Not used	
35	SAT (20)		Not used	
36	SDA (1)	STATN (1)	Latitude, first station	Identification and position information for up to 10 stations (rad, rad, Km, -)
37	SDA (2)	STATN (2)	Longitude, first station	
38	SDA (3)	STATN (3)	Altitude, first station	
39	SDA (4)	STATN (4)	Station name, first station	
40	SDA (5)	STATN (5)	Latitude, second station	
⋮	⋮	⋮	⋮	
75	SDA (40)	STATN (40)	Station name, tenth station	
76	SDA (41)	HØRCØR (1)	First station	Horizon corrections for each station (rad)
⋮	⋮	⋮	⋮	
85	SDA (50)	HØRCØR (10)	Tenth station	
86	SDA (51)	STERR (1)	Latitude variance, first station	Position error for each station (rad^2, rad^2, Km^2)
87	SDA (52)	STERR (2)	Longitude variance, first station	
88	SDA (53)	STERR (3)	Altitude variance, first station	
89	SDA (54)	STERR (4)	Latitude variance, second station	
90	SDA (55)	STERR (5)	Longitude variance, second station	
⋮	⋮	⋮	⋮	
115	SDA (80)	STERR (30)	Altitude variance, tenth station	
116	SDA (81)	STERR (31)	Not used	
⋮	⋮	⋮	⋮	
175	SDA (140)	STERR (90)	Not used	

Array DATA Location	Subarray	FØRTRAN Names	Description
176	SDA (141)	SNØISE (1)	Range variance, first station
177	SDA (142)	SNØISE (2)	Range rate variance, first station
178	SDA (143)	SNØISE (3)	Azimuth variance first station
179	SDA (144)	SNØISE (4)	Elevation variance first station
180	SDA (145)	SNØISE (5)	Range variance, second station
⋮	⋮	⋮	⋮
215	SDA (180)	SNØISE (40)	Elevation variance, tenth station
216	SDA (181)	SNØISE (41)	Not used
⋮	⋮	⋮	⋮
275	SDA (240)	SNØISE (100)	Not used
276	SDA (241)	NUMBER	Number of tracking stations employed
277			Not used
⋮			⋮
285			Not used
286	STT (1)	PHI (1,1)	Transition matrix relating errors at two successive data points. WRK (42, 43) to WRK (61,62)
⋮	⋮	⋮	
321	STT (36)	PHI (6,6)	
322	STT (37)	ØBST (1,1)	Matrix of partials of observations with respect to the state at time WRK (61,62)
⋮	⋮	⋮	
339	STT (54)	ØBST (6,3)	
340	STT (55)	Q (1,1)	Covariance matrix for contribution of errors in observations and of errors in station location at the time WRK (61,62)
⋮	⋮	⋮	
348	STT (63)	Q (3,3)	
349	STT (64)	STATE (1)	Vector of position – velocity differentials at the time WRK (42, 43) Km, Km/sec.
⋮	⋮	⋮	
354	STT (69)	STATE (6)	
355	STT (70)	PP (1,1), P (1,1)	Covariance matrix for errors in the estimates of the state vector at time WRK (61,62)
⋮	⋮	⋮	
390	STT (105)	PP (6,6), P (6,6)	

Measurement errors for each station (Km^2, Km^2/sec^2, rad^2, rad^2)

DATA Location	Subarray	FØRTRAN Names	Description
391	WRK (1)	RØTATE (1,1), AN (1,1)	Transformation matrix, and its inverse, relating true equator of date frame of reference to the mean equinox of 1950.0 system ($\vec{r}_D$ = ROTATE $\vec{r}_{so}$)
⋮	⋮	⋮	
399	WRK (9)	RØTATE (3,3), AN (3,3)	
400	WRK (10)	RØTINV (1,1)	
⋮	⋮	⋮	
408	WRK (18)	RØTINV (3,3)	
409	WRK (19)	EN (1,1)	Nutation array relating the mean equator of date to the true equator of date ($\vec{r}_D$ = EN $\vec{r}_M$)
⋮	⋮	⋮	
417	WRK (27)	EN (3,3)	
418	WRK (28)	RCØN(1),RCØ(1),RCØNIC(1)	Position and velocity vectors used to describe the conic reference trajectory (frame of 1950.0) at the most recent rectification. Epoch defined by WRK (34, 35) (Km, Km/sec)
419	WRK (29)	RCØN(2),RCØ(2),RCØNIC(2)	
420	WRK (30)	RCØN(3),RCØ(3),RCØNIC(3)	
421	WRK (31)	VCØN(1),VCØ(1),VCØNIC(1)	
422	WRK (32)	VCØN(2),VCØ(2),VCØNIC(2)	
423	WRK (33)	VCØN(3),VCØ(3),VCØNIC(3)	
424	WRK (34)	TCØNW, TWCØN	Whole day — Epoch corresponding to the radius and velocity vectors used to define the reference ellipse (days)
425	WRK (35)	TCØNF, TFCØN	Fractional day
426	WRK (36)	RTRAN(1), RTRANS(1)	Position and velocity vectors utilized to define the state from which errors will propagate to the time of data acquisition. These vectors are in the true equator of data frame (Km, Km/sec)
427	WRK (37)	RTRAN(2), RTRANS(2)	
428	WRK (38)	RTRAN(3), RTRANS(3)	
429	WRK (39)	VTRAN(1), VTRANS(1)	
430	WRK (40)	VTRAN(2), VTRANS(2)	
431	WRK (41)	VTRAN(3), VTRANS(3)	

DATA Location	Array Subarray	FØRTRAN Names	Description
432	WRK (42)	TTRANW	Whole day } The epoch defining the point from which errors will be propogated (days)
433	WRK (43)	TTRANF	Fractional day
434	WRK (44)	R50(1),RVEC(1),R(1), RAD(1),RA(1)	Satellite position and velocity vectors in the frame of 1950.0 corresponding initially to SAT (5) and SAT (8) at the epoch defined by WRK (50,51) (Km, Km/sec.)
435	WRK (45)	R50(2),RVEC(2),R(2), RAD(2),RA(2)	
436	WRK (46)	R50(3),RVEC(3),R(3), RAD(3),RA(3)	
437	WRK (47)	V50(1),VVEC(1),V(1), VEL(1),VE(1)	
438	WRK (48)	V50(2),VVEC(2),V(2), VEL(2),VE(2)	
439	WRK (49)	V50(3) VVEC(3),V(3), VEL(3),VE(3)	
440	WRK (50)	TW	Whole day } Days from J. D. 2433283.423 (1950.0) (Instanteous record of time)
441	WRK (51)	TF	Fractional day
442	WRK (52)	DR(1),R(1),DELTA(1), X(1)	Instantaneous satellite position vector relative to the reference conic, in the frame of 1950.0. (Km)
443	WRK (53)	DR(2),R(2),DELTA(2), X(2)	
444	WRK (54)	DR(3),R(3),DELTA(3), X(3)	
445	WRK (55)	DV(1),RD(1),DRDØT(1), XD(1)	Instantaneous satellite velocity and acceleration vectors, relative to the reference conic, in the frame of 1950.0. (Km/sec, Km/sec^2)
446	WRK (56)	DV(2),RD(2),DRDØT(2), XD(2)	
447	WRK (57)	DV(3),RD(3),DRDØT(3), XD(3)	
448	WRK (58)	RDD(1),XDD(1)	
449	WRK (59)	RDD(2),XDD(2)	
450	WRK (60)	RDD(3),XDD(3)	

DATA Location	Array Subarray	FØRTRAN Names	Description
451	WRK (61)	TSTØPW,TWDATA,TW	Time of next observed data points relative to 1950.0.
452	WRK (62)	TSTØPF,TFDATA,TF	
453	WRK (63)	ITRAK,ISTN	Station which next senses the vehicle
454	WRK (64)	ITYPE	Indicates type of observed data being processed
455	WRK (65)	ØDATA(1)	Observed values of range, range rate, elevation and azimuth as specified by WRK (64). (Km, Km/sec., rad, rad) See subroutine FILTER
456	WRK (66)	ØDATA(2)	
457	WRK (67)	ØDATA(3)	
458	WRK (68)	H.STEP	Stepsize for the numerical integration. (sec)
459	WRK (69)	TW PRNT, TQ	Days from JD 2433282.5.
460	WRK (70)	TF PRNT, TP	U.T. for day beginning TW PRINT
461	WRK (71)	CØEF(1,1)	$\Sigma^2 \ddot{x}$
462	WRK (72)	CØEF(2,1)	$\Sigma' \ddot{x}$ Numerical sums and differences evaluated along a trailing diagonal for a Gauss-Jackson stepwise integration of the equations of motion
463	WRK (73)	CØEF(3,1)	$\Delta' \ddot{x}$
464	WRK (74)	CØEF(4,1)	.
465	WRK (75)	CØEF(5,1)	.
466	WRK (76)	CØEF(6,1)	.
467	WRK (77)	CØEF(7,1)	
468	WRK (78)	CØEF(8,1)	$\Delta^6 \ddot{x}$
469	WRK (79)	CØEF(1,2)	$\Sigma^2 \ddot{y}$
⋮	⋮	⋮	⋮
484	WRK (94)	CØEF(8,3)	$\Delta^6 \ddot{z}$
485	WRK (95)	RT(1),RTRAK(1)	Position vector of tracking station in the true equator frame of date. (Km)
486	WRK (96)	RT(2),RTRAK(2)	
487	WRK (97)	RT(3),RTRAK(3)	
488	WRK (98)	RSUN(1),RS(1)	Radius vector of the sun in the frame of 1950.0 relative to the Earth. Epoch defined by WRK (50,51)
489	WRK (99)	RSUN(2),RS(2)	
490	WRK (100)	RSUN(3),RS(3)	
491	WRK (101)	RM(1)	Radius vector of the moon in the frame of 1950.0 relative to the Earth. Epoch defined by WRK (50,51)
492	WRK (102)	RM(2)	
493	WRK (103)	RM(3)	
494	WRK (104)	TIME	Time interval since the reference trajectory was last rectified (sec)

DATA Location	Array Subarray	FØRTRAN Names	Description
495	WRK (105)	SLAT	Tracking station latitude (Geodetic) longitude and altitude (rad, rad, Km)
496	WRK (106)	SLØN	
497	WRK (107)	SALT, H	
498	WRK (108)	U(1),X(1)	Up unit vector at the tracking station being checked. True equator of date frame of reference.
499	WRK (109)	U(2),X(2)	
500	WRK (110)	U(3),X(3)	
501	WRK (111)	RDATE(1),RVEC(1)	Instantaneous position and velocity vectors in the true equator of date frame (Km, Km/sec)
502	WRK (112)	RDATE(2),RVEC(2)	
503	WRK (113)	RDATE(3),RVEC(3)	
504	WRK (114)	VDATE(1),VVEC(1)	
505	WRK (115)	VDATE(2),VVEC(2)	
506	WRK (116)	VDATE(3),VVEC(3)	
507	WRK (117)	RCØNIC(1),RC(1)	Position and velocity vectors on the conic reference trajectory at the epoch (WRK (50,51))
508	WRK (118)	RCØNIC(2),RC(2)	
509	WRK (119)	RCØNIC(3),RC(3)	
510	WRK (120)	VCØNIC(1),VC(1)	
511	WRK (121)	VCØNIC(2),VC(2)	
512	WRK (122)	VCØNIC(3),VC(3)	
513	WRK (123)	HRK,H	Step size employed by the Runge-Kutta starter (= $\frac{1}{2}$ of WRK (68))
514	WRK (124)	E(1),Y(1)	East unit vector at the tracking station being checked. True equator of date frame.
515	WRK (125)	E(2),Y(2)	
516	WRK (126)	E(3),Y(3)	
517	WRK (127)	Z(1)	North unit vector at the tracking station being checked. True equator of date frame.
518	WRK (128)	Z(2)	
519	WRK (129)	Z(3)	
520	WRK (130)	ELEM(1),A(1)	Array of constants utilized to define the conic reference trajectory
521	WRK (131)	ELEM(2),A(2)	
522	WRK (132)	ELEM(3),A(3)	
523	WRK (133)	ELEM(4),A(4)	
524	WRK (134)	NSTEPS, N	Number of integration steps from one data point to the next.
525	WRK (135)		Not used

Labeled CØMMØN MAP

In addition to blank CØMMØN, CØMMØN storage utilized by the program includes three regions of labeled CØMMØN. These latter regions, containing tabulated atmospheric and ephemeris data, are set at load time by means of a BLØCK DATA subprogram. These regions are explicitly accessed by subroutines ATMS and EPHEM exclusively. A description of the data stored in the labeled CØMMØN regions is tabulated below.

Labeled CØMMØN MAP

Location	FØRTRAN Symbol	Units	FØRTRAN Dimension	Description
ATCØN(1)	ALT1	km	1	Altitude limits for density data.
ATCØN(2)	ALT2	km	1	
ATCØN(3)	ALT3	km	1	
ATCØN(4)	REQT	km	1	Equatorial radii of Earth
ATCØN(5)	RPØL	km	1	Polar radii of Earth
ATCØN(6)	STEP1	km	1	Interval between ALT1 and ALT2
ATCØN(7)	STEP2	km	1	Interval between ALT2 and ALT3
TABLE(1)	RHØF		21	Density for 1962 U. S. Standard atmosphere at tabulated intervals specified by ATCØN.
TABLE(22)	RATE		21	Lapse rate for 1962 U. S. Standard atmosphere at tabulated intervals specified by ATCØN.
EPHØM(1)	EPHØM(1)	days	1	Date relative to 1950.0 at which the tabulated emphemeris data begin
EPHØM(2)	EPHØM(2)	days	1	Time interval for solar ephemeris entries
EPHØM(3)	EPHØM(3)	days	1	Time interval for lunar ephemeris entries
EPHØM(4) ⋮ EPHØM(273)	EPHØM(4) ⋮ EPHØM(273)	km, km/sec	270	Lunar position and velocity vectors. Ordered X_1, Y_1, Z_1, V_{x_1}, V_{y_1}, V_{z_1}, X_2, Y_2 ...
EPHØM(274) ⋮ EPHØM(339)	EPHØM(274) ⋮ EPHØM(339)	km, km/sec	63	Solar position and velocity vectors. Ordered X_1, Y_1, Z_1, V_{x_1}, V_{y_1}, V_{z_1}, X_2, Y_2, ...

APPENDIX II

PROGRAM INPUT/OUTPUT TAPES

The program logic has been constructed in such a manner that variable input/output tape numbers can be employed so that operation on any system with a basic FORTRAN capability will be possible with a minimum of effort. This mode of operation has been provided by placing the desired tape numbers in COMMON [CON(16), CON(17)] while loading subroutine INPUT and calling for them in each routine requiring input or output.

Unfortunately, due to problems of checkout with the present NAASYS system, it was necessary to revise this logic and insert the actual tape units utilized in each routine.

INPUT	tape unit 5
OUTPUT	tape unit 6
SPECIAL	tape unit 9

The linkage to COMMON was not, however, removed. Thus, if sufficient CORE exists to allow communication with each tape unit in this mode, the capability should be reinstated to assure complete compatibility of the program with all FORTRAN systems and to avoid problems associated with updating or changing the system.

For the sake of reference, the locations of all input/output statements in the program are listed below.

SUBROUTINE	CARDS
MAIN	0390
INPUT	3830, 3910, 3931
REED	0280, 0410
SICRD	0026, 0058
DADUMP	0210, 0250, 0330, 0370, 0450, 0490, 0500, 0580, 0620, 0700, 0740, 0795
ATMS	0340
TRAK	0910, 1022, 1480, 1530, 1550
UPSTAT	0350, 0370, 0400, 0420, 0660, 0770
DATAPE	0690, 0720, 0880
ATANS	0270
CHOOSE	0160

The preceding comments pertain only to those systems for which the use of control cards (in the binary deck) to equivalence the addresses of the tape units requested and those actually utilized is not permitted. However, if such operation is possible, this "fix" is by far the simplest remedy.

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) North American Aviation, Inc. Space and Information Systems Division Downey, California	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP N/A

3. REPORT TITLE

Orbit Differential Correction - Tracking Program
Differential Correction Geocentric Orbit Computations Program

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Final Report

5. AUTHOR(S) (Last name, first name, initial)

Townsend, George E.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
April 1966	577	

8a. CONTRACT OR GRANT NO. AF30(602)-3638 b. PROJECT NO. 4519 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) SID 65-1203-1, -2, -3 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) RADC-TR-66-67, Volume I of III Volumes

10. AVAILABILITY/LIMITATION NOTICES

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Rome Air Development Center (EMCRR) Griffiss Air Force Base, New York 13440

13. ABSTRACT

This document presents the formulation, computational logic and coding information developed for the purpose of effecting the definition of geocentric satellite orbits. The rationale for this process is constructed around the recursive minimum variance data filter developed by R. E. Kalman and a specially prepared magnetic tape generated in the preprocessor (SID 65 1203-2).

The trajectory portion of the program is formulated in the Encke manner and includes perturbing accelerations resulting from the first 3 harmonics of the Earth's potential function, atmospheric drag, solar radiation pressure, and solar and lunar gravitation. These accelerations are integrated via an uncorrected Gauss-Jackson routine started with a fourth order Runge-Kutta process.

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A ROLE	LINK A WT	LINK B ROLE	LINK B WT	LINK C ROLE	LINK C WT
Orbit analysis, orbit differential correction, satellite tracking program						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2*a*. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2*b*. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7*a*. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7*b*. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8*a*. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8*b*, 8*c*, & 8*d*. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9*a*. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9*b*. OTHER REPORT NUMBER(S). If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through ________________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through ________________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through ________________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY. Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.

UNCLASSIFIED

Security Classification